

09/3416336

PCT/JP98/00233

22.01.98

日本国特許庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application:

1997年 1月23日

REC'D 20 MAR 1993

WIPO PCT

出願番号
Application Number:

平成 9年特許願第010592号

出願人
Applicant(s):

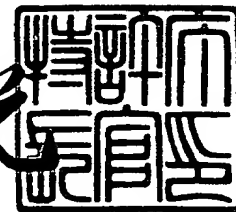
シャープ株式会社

PRIORITY DOCUMENT

1998年 3月 6日

特許庁長官
Commissioner,
Patent Office

荒井寿光



出証番号 出証特平10-3012647

【書類名】 特許願

【整理番号】 96-01454

【提出日】 平成 9年 1月23日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 3/14
G06F 3/153

【発明の名称】 プログラマブル表示装置

【請求項の数】 9

【発明者】

【住所又は居所】 大阪府大阪市阿倍野区長池町22番22号 シャープ株式会社内

【氏名】 中村 聡

【発明者】

【住所又は居所】 大阪府大阪市阿倍野区長池町22番22号 シャープ株式会社内

【氏名】 山村 博幸

【発明者】

【住所又は居所】 大阪府大阪市阿倍野区長池町22番22号 シャープ株式会社内

【氏名】 山本 真司

【発明者】

【住所又は居所】 大阪府大阪市阿倍野区長池町22番22号 シャープ株式会社内

【氏名】 守屋 政明

【特許出願人】

【識別番号】 000005049

【郵便番号】 545

【住所又は居所】 大阪府大阪市阿倍野区長池町22番22号

【氏名又は名称】 シャープ株式会社

【代表者】 辻 晴雄

【代理人】

【識別番号】 100069534

【弁理士】

【氏名又は名称】 藤本 博光

【電話番号】 03(3593)2361

【手数料の表示】

【予納台帳番号】 007951

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9106003

【書類名】 明細書

【発明の名称】 プログラマブル表示装置

【特許請求の範囲】

【請求項1】 表示データが格納されているメインメモリと、

前記表示データのデータ形式を画面表示のデータ形式に変換するデータ処理回路部と、

前記データ処理回路部に変換された表示データを表示ライン単位に格納する複数のラインメモリと、

前記メインメモリから前記ラインメモリに表示データを転送格納させて、該ラインメモリから必要な表示データを読み出して画面表示させる制御を行う表示制御部と、

前記メインメモリに前記表示データを格納させ、データ形式及び格納アドレスを含む格納情報を前記表示制御部に転送する主制御部と、
を備え、

前記表示制御部は、転送元の前記メインメモリに対し前記格納情報に基づいて、画面表示させる可能性のある1ライン分の表示データのアドレスを指定して該表示データを読み出し、前記データ処理回路部にデータ変換をさせて、前記ラインメモリを選択し該表示データを格納させることを特徴とするプログラマブル表示装置。

【請求項2】 前記表示制御部は、前記ラインメモリに繰り返し利用する表示データを格納し、該繰り返し表示データを表示させる場合には、前記ラインメモリから前記繰り返し表示データのアドレスを指定して読み出し、画面表示させることを特徴とする請求項1記載のプログラマブル表示装置。

【請求項3】 繰り返し利用する表示データを格納するデータバッファメモリを備え、

前記表示制御部は、前記データを画面表示させる場合、前記データバッファメモリから前記繰り返し表示データを読み出し、画面表示させることを特徴とする請求項1記載のプログラマブル表示装置。

【請求項4】 前記メインメモリから読み出した表示データを格納する第1バッファメモリと、

前記第1バッファメモリから読み出した表示データを格納する第2バッファメモリと、

前記第1及び第2バッファメモリの読み出し及び書き込みアドレスをカウントするアドレスカウンタと、

を備え、

前記表示制御部は、前記アドレスカウンタに対し読み出し及び書き込みアドレスカウントをそれぞれ停止／動作の制御を行い、拡大・縮小・スキップの処理を行って、そのデータを前記ラインメモリに格納することを特徴とする請求項1記載のプログラマブル表示装置。

【請求項5】 前記表示制御部は、第1バッファメモリからの読みだしアドレスカウントを所定の順に停止／動作を繰り返させることを特徴とする請求項4記載のプログラマブル表示装置。

【請求項6】 前記データ処理回路部は、各種のデータ形式を変換する複数の変換処理回路を有し、

前記表示制御部は、前記格納情報のデータ形式情報に基づいて前記変換処理回路を選択することを特徴とする請求項1記載のプログラマブル表示装置。

【請求項7】 前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えたことを特徴とする請求項1記載のプログラマブル表示装置。

【請求項8】 前記表示制御部は、前記プログラムメモリと前記データメモリに必要な情報を前記メインメモリから転送させることを特徴とする請求項7記載のプログラマブル表示装置。

【請求項9】 前記表示制御部は、前記ラインメモリに表示データを格納する際に何ライン目で使用するデータであるかを示すライン情報を付加し、前記ラインメモリから表示データを読み出す際にライン情報も同時に読み出して、該表示データを使用するラインがライン情報と同一である場合のみ画面表示させることを特徴とする請求項1記載のプログラマブル表示装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は映像データを表示するコンピュータ装置におけるプログラマブル表示装置に関し、特にグラフィックディスプレイシステムにおけるメモリからの表示用データの読み出し方が極めて柔軟であり、メモリから表示用データを読み出す際に読み出す画素データの最小単位を画素毎として、動的に定義できるシステムに関する。

【0002】

【従来の技術】

従来、一般的なコンピュータにおいて、表示データの重ね合わせや合成処理は、単一のフレームメモリ上において、そのメモリ上で直接メインプロセッサあるいは描画装置が演算して合成する。図27は、従来の画像表示装置の一例を示すブロック図である。この画像表示装置は、メインCPU101、メインメモリ102、データ処理回路103、ラインメモリ104、出力処理回路105、システムコントローラ106、同期信号生成回路107からなる構成である。

【0003】

メインメモリ102には、表示データがいくつか格納されている。例えば、数種類のウィンドウ表示を行う場合を考えると、各ウィンドウに対応した表示データが格納されている。このウィンドウを重ね合わせて一画面に表示させる場合、一画面表示になるように、メインCPU101が各表示データを選択して読み出し、一画面の表示データを再びメインメモリ102に格納する。同期信号生成回路107で発生した同期信号のタイミングに従って、システムコントローラ106がデータ転送用にメインメモリ102のアドレスを生成する。このアドレスに従ってメインメモリ102から表示データを読み出して、あらかじめ決められたデータ処理をデータ処理回路103で行った後、ラインメモリ104にデータを転送する。ラインメモリ104からのデータは同期信号のタイミングにしたがって出力され、出力処理回路105により表示用の処理を行ってディスプレイに表示する。

【0004】

また、特開平6-149527号公報に開示されているように、重ねあわせに必要な枚数分フレームメモリを用意して、映像出力時に全部のフレームメモリからデータを読み出し各フレーム間の優先順位を元に合成結果を表示するシステムがある。

【0005】

また、特開平6-295169号公報に開示されているように、表示用のメモリとは別に表示領域のメモリの各表示ドット毎に識別メモリを設けることで各表示ドットが今どのようなモード（例えば1画素のビット数）にあるかを識別し、そのモードに合わせて表示し、1つの画面上で異なる表示モードを表示するシステムがある。

【0006】

また、前記特許にもあるように識別メモリの内容を参照したり、特開平7-334342号公報に開示するように別途マスクメモリを利用し、表示している各ウインドウ内の情報を変更して書き換える場合にその領域外をマスクするシステムがある。

【0007】

【発明が解決しようとする課題】

しかしながら従来のようにメインCPU101が、各ウインドウの重ね合わせ等の処理を行う場合には、メインCPU101の負担が過大となり、他の処理を行うことができずに全体の処理速度が遅くなるなどの問題があった。

【0008】

また、各ウインドウの重ねあわせをするのに必要な枚数分のフレームメモリを持つことで、ソフトウェアの処理負荷を軽減する方法では、始めからそのシステムで必要と思われる最大枚数分のフレームメモリが必要である。つまり、画面上に表示するウインドウのサイズに関係なく表示エリアの最大サイズでフレームメモリを必要とする。そのため、メモリの利用効率が極めて悪くなる上に、多数のウインドウを同時に開いた場合、そのウインドウに対応する全てのフレームメモリから同時にデータを読み出す必要がある。すなわち、ウインドウが重なってお

り実際は表示されないような部分のデータも読み出す必要がある。こうして、画面上に開くウィンドウの枚数に比例して消費電力が大きくなってしまふ。

【0009】

また、従来の様に1つの画面上で異なる表示モードを混在表示する方法として表示領域のメモリの各表示ドット毎に識別メモリを設けることで各表示ドットが今どのようなモードにあるかを識別する方法がある。その方法では、フルスクリーン分のメモリに対して別途数ビットの識別メモリが必要となるため、別の用途に転用することのできないメモリ（識別メモリ）が余分に必要となる。このことは、マスクメモリを利用する場合にも同様なことが言える。

【0010】

本発明の目的は、表示データを収納するメモリ空間を必要なものだけとし、表示のためのメモリアクセス回数を押さえて処理を高速化できるとともに、主制御部の負担を軽減することができるプログラマブル表示装置を提供することである。

【0011】

【課題を解決するための手段】

請求項1の発明は、表示データが格納されているメインメモリと、前記表示データのデータ形式を画面表示のデータ形式に変換するデータ処理回路部と、前記データ処理回路部に変換された表示データを表示ライン単位に格納する複数のラインメモリと、前記メインメモリから前記ラインメモリに表示データを転送格納させて、該ラインメモリから必要な表示データを読み出して画面表示させる制御を行う表示制御部と、前記メインメモリに前記表示データを格納させ、データ形式及び格納アドレスを含む格納情報を前記表示制御部に転送する主制御部と、を備えたプログラマブル表示装置である。前記表示制御部は、転送元の前記メインメモリに対し前記格納情報に基づいて、画面表示させる可能性のある1ライン分の表示データのアドレスを指定して該表示データを読み出し、前記データ処理回路部にデータ変換をさせて、前記ラインメモリを選択し該表示データを格納させることを特徴とする。

【0012】

請求項2の発明は、前記表示制御部は、前記ラインメモリに繰り返し利用する表示データを格納し、該繰り返し表示データを表示させる場合には、前記ラインメモリから前記繰り返し表示データのアドレスを指定して読み出し、画面表示させることを特徴とする請求項1記載のプログラマブル表示装置である。

【0013】

請求項3の発明は、繰り返し利用する表示データを格納するデータバッファメモリを備え、前記表示制御部は、前記データを画面表示させる場合、前記データバッファメモリから前記繰り返し表示データを読み出し、画面表示させることを特徴とする請求項1記載のプログラマブル表示装置である。

【0014】

請求項4の発明は、前記メインメモリから読み出した表示データを格納する第1バッファメモリと、前記第1バッファメモリから読み出した表示データを格納する第2バッファメモリと、前記第1及び第2バッファメモリの読み出し及び書き込みアドレスをカウントするアドレスカウンタとを備える請求項1記載のプログラマブル表示装置である。前記表示制御部は、前記アドレスカウンタに対し読み出し及び書き込みアドレスカウントをそれぞれ停止／動作の制御を行い、拡大・縮小・スキップの処理を行って、そのデータを前記ラインメモリに格納することを特徴とする。

【0015】

請求項5の発明は、前記表示制御部が、データ第1バッファメモリからの読みだしアドレスカウントを所定の順に停止／動作を繰り返すことを特徴とする請求項4記載のプログラマブル表示装置である。

【0016】

請求項6の発明は、前記データ処理回路部は、各種のデータ形式を変換する複数の変換処理回路を有し、前記表示制御部は、前記格納情報のデータ形式情報に基づいて前記変換処理回路を選択することを特徴とする請求項1記載のプログラマブル表示装置である。

【0017】

請求項7の発明は、前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えたことを特徴とする請求項1記載のプログラマブル表示装置である。

【0018】

請求項8の発明は、前記表示制御部は、前記プログラムメモリと前記データメモリに必要な情報を前記メインメモリから転送させることを特徴とする請求項7記載のプログラマブル表示装置である。

【0019】

請求項9の発明は、前記表示制御部は、前記ラインメモリに表示データを格納する際に何ライン目で使用するデータであるかを示すライン情報を付加し、前記ラインメモリから表示データを読み出す際にライン情報も同時に読み出して、該表示データを使用するラインがライン情報と同一である場合のみ画面表示させることを特徴とする請求項1記載のプログラマブル表示装置である。

【0020】

請求項1の発明において、表示をする際に必要な部分の表示データをメインメモリ内から取り出して使用する。そのため、メインメモリ内の任意の位置のデータを取り出して任意に組み合わせることが可能である。この制御はすべて表示制御部が行い、主制御部が処理を行う必要がなく、主制御部のソフトウェアにおける処理負荷を低減できる。

【0021】

請求項2の発明において、ウインドウシステムにおける背景等のように、ライン方向に対して繰り返すようなデータであった場合、読み出しラインメモリアドレスを任意の位置でループできる。

【0022】

請求項3の発明において、カーソルや繰り返し背景などをデータバッファメモリに収納しておけるため、決まりきったデータをメインメモリから読み出す必要が無い。データバスの使用回数を減らすことができる。

【0023】

請求項4の発明において、表示データを読み出す際に拡大縮小処理をするため、表示用データに対する拡大縮小処理を事前にする必要が無く、バスの使用効率を上げられる。また、ビデオ入力映像を表示する場合に映像サイズの変更が必要となるのが常であるが、出力段に拡大縮小処理を掛けることで拡大縮小回路がより有効に利用できる。また、この事によりビデオデータを常にフルサイズで取り込みながら、そのデータを一旦フレームメモリなどに転送することなく表示は任意のサイズに設定できる。

【0024】

請求項5の発明において、第1バッファメモリからの読みだしアドレスカウンタを所定の順に停止/動作を繰り返すことにより、一定倍率の拡大・縮小が簡単な処理で行うことができる。

【0025】

請求項6の発明において、表示制御部は、格納情報のデータ形式情報に基づいてデータ変換ができるので、表示用データを収納する形式などに制限が無い。

【0026】

請求項7の発明において、前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えるので、処理の度にメインメモリからデータを読み出す必要がない。

【0027】

請求項8の発明において、前記表示制御部は、前記プログラムメモリとデータメモリに必要な情報をメインメモリから転送させるため、画面モードあるいはグラフィック領域の変更に柔軟に対応できる。容量を越えたプログラムあるいはデータは、メインメモリから読み出せばよいので、メモリの容量は小さくて済む。

【0028】

請求項9の発明において、各ラインの表示毎にラインメモリ内のデータを消去する必要がなく、垂直帰線期間毎にすべてのラインメモリの使用ライン情報を消去するだけでよいので処理の高速化を図れる。

【0029】

【発明の実施の形態】

以下に、本発明の実施形態について図面を用いて説明する。

図1は、本発明に係るプログラマブル表示装置の一実施形態を示すブロック図である。この表示装置は、メインCPU11、プログラムや表示データやその他のデータを記憶するメインメモリ12、メインメモリ12の表示データをディスプレイ表示のデータ形式に変換する処理を行うデータ処理回路13、変換処理された表示データを記憶する表示メモリ部14、表示データを画面に出力するための処理を行う出力処理回路17、メインメモリ12へのデータアクセスを行うDMA (Direct Memory Access) 18、プログラムメモリ19、データメモリ20、プログラムメモリ19やデータメモリ20に記述された命令・データを解釈し、それに従っておもに表示データの転送等を行う表示プロセッサ21、同期信号生成回路22、ビデオ入力23、24とから構成される。

【0030】

データ処理回路13は、図2に示すように表示プロセッサ19より送られてきた表示データに対してYUV→RGB変換を行うYUVデコーダ27a、同表示データに対してランレングス展開を行うランレングス展開回路27b、同表示データに対してカラーデータの伸長を行うカラー伸長回路27c、同表示データに対してパレット変換を行う複数のカラーパレット27d、27eの複数の処理回路と、セレクタ28とからなる。表示メモリ部14は、図2に示すように、カーソルのパターンデータなどの格納に使用できるデータバッファ15と、データ表示データおよび使用ライン情報を記憶する複数のラインメモリ16とからなる。出力処理回路17は、複数のラインメモリ16より任意のラインメモリを選択するセレクタ、 α ブレンディングを実現するため表示データの明るさを変化させるアッテネータおよびその出力を加算する加算器、繰り返し背景データやカーソルなどの合成に使用されるセレクタ、ディスプレイに表示するためD/A変換を行うD/Aコンバータ等からなる。表示プロセッサ21は、図3に示すように、転送用バッファメモリ25a、25b、26a、26bを有する。

【0031】

この表示装置は、専用のフレームバッファを持たず、メインメモリ12に表示データを同居させるUMA (Unified Memory Architecture)構成を取り入れるが、メインメモリ12に専用のフレームバッファを持つ構成になってもかまわない。

【0032】

以下、この実施形態の動作を説明する。

まず、表示データが実際に表示されるまでのおおまかな流れを以下に説明する。メインCPU11により表示データは主にメインメモリ12に格納されている。これらの表示データは、DMA18によって読み出され、図3に示す表示プロセッサ21内部の転送用バッファメモリ25a, 25bに一時的に格納される。そこで拡大・縮小・スキップなどの操作をされて転送用バッファメモリ26a, 26bに格納されたのち、データ処理回路13により単純なRGB形式のデータに変換され、ラインメモリ16に格納される。ラインメモリ16に書き込まれたデータは、同期信号生成回路22が発生する同期信号のドットクロックに合わせて1画素分ずつ読み出される。出力処理回路17によって2画面の α ブレンディング処理、又は繰り返し背景データやカーソルなどと合成され、D/A変換されて同期信号などと共にディスプレイに出力されて表示される。以上が表示までのおおまかな流れである。

【0033】

この表示装置において、表示のかなりの部分の制御は表示プロセッサ21よって行われる。表示プロセッサ21は専用のプログラムメモリ19およびデータメモリ20を持っており、それに格納されているプログラムおよびデータを解釈して、それに従って表示データの転送等を行う。プログラムメモリ19およびデータメモリ20の情報は、必要に応じてメインメモリ12から転送されてくる。メインメモリ12には表示構成、グラフィック領域の変更などに応じて複数のプログラム/データを格納しておく。

【0034】

メインメモリ12からの表示データの転送命令は、メインCPU11から直接

表示プロセッサ21に対し発行する場合と、表示プロセッサ21自身が発行する場合がある。転送命令をメインCPU11が発行するのは、主に表示モード（1画素の情報を示すビット数）が変更になった場合であり、表示プロセッサ21自身が発行するのは、主に1画面を構成するために必要なプログラム／データが表示プロセッサのRAM容量より大きい場合である。このとき表示の途中でプログラム／データを入れ換える。

【0035】

このような構成にすることで、必要に応じて情報を転送すればよいと、画面モードあるいはグラフィック領域の変更に柔軟に対応できる。かつメモリ容量を越えたプログラムあるいはデータを実行可能とする。こうして表示プロセッサのRAMは小容量で済む。コンパクトあるいは低コストでシステムを構築できる。

【0036】

また表示プロセッサ21の動作が決まってい変更が必要ない場合は、プログラムメモリ19またはデータメモリ20はROMでもかまわない。この場合は、メインメモリ12から転送するには及ばない。ROMは同容量のRAMに比べてチップ面積が小さくて済むため、コスト面で有利になる。

【0037】

次に、表示プロセッサ21にプログラムを与えて画面表示を行うときの、表示プロセッサ21の基本動作について説明する。図4（A）～（C）は、メインメモリ12の表示データとディスプレイの表示出力を示す説明図である。いずれもあらかじめメインメモリ12上に格納してある表示データをラインメモリ16に格納するものである。ベタスクリーンを表示する場合と複数のウィンドウ等を合成して表示する場合について説明する。

【0038】

ベタスクリーンとは、図4（A）に示すようにメインCPU11により、背景、カーソル、ウィンドウ等を、合成されたベタスクリーンデータとして、あらかじめメインメモリ12上に格納しておく。表示するには格納されている先頭アドレスから順に読み出してラインメモリ16に転送していけばよい。

【0039】

複数ウィンドウ等の合成には、 α ブレンディングを考慮しない場合（図4（B）参照）と、考慮する場合（図4（C）参照）の2つの場合がある。 α ブレンディングとは半透明合成のことで、例えば2つのウィンドウが重なる場合、通常ならば重なった部分は手前のウィンドウだけが表示されるが、 α ブレンディングの指定をすると手前のウィンドウが透けて、奥のウィンドウが見えるようになる。言い換えると、 α ブレンディングとは、このように複数の表示データのある割合で合成して表示する機能のことを言う。一方、表示プロセッサ21の動作には実際には拡大・縮小・スキップの操作やデータ処理回路13や使用ライン情報などの制御が関わってくるが、それらの動作については後述する。

【0040】

次に、これら画面表示を行う場合の表示装置の動作について説明する。図5は、ベタスクリーンデータを1画面分表示するフローチャートである。図6はそのときの表示画面例であり、図7はベタスクリーンデータが格納されているメインメモリ12のメモリマップである。まず、1画面を表示するために、ステップA1において、ベタスクリーンデータのX方向の座標サイズ x_1 、ベタスクリーンデータのY方向の座標サイズ y_1 を取得する。

【0041】

次に、ステップA2にて、メインメモリ12上のベタスクリーンデータ格納先頭アドレス`beta_addr`をライン番号Lに相当するベタスクリーン格納先頭アドレス`addr`として取得する。これらのデータは、固定データであればプログラム内で固定されたイミディエイトデータとして取得することもできる。また、任意のサイズであればメインメモリ12上に存在するこれらのデータをデータメモリ20に転送し、データメモリ20を参照することによっても取得することが可能である。ステップA3にて、現在表示中の水平ライン番号の次の水平ライン番号をライン番号Lとして取得し、この値が偶数か奇数かをステップA4にて判別する。偶数であれば、ステップA5にてラインメモリ16aへ、奇数であれば、ステップA6にてラインメモリ16bへ、メインメモリ12のライン番号Lに相当するベタスクリーンデータ格納先頭アドレス`addr`から x_1 サイズの

データ転送を行う。

【0042】

偶数と奇数のライン番号によりラインメモリ16aとラインメモリ16bへの書き込みを切り替えているのは、ラインメモリが表示側でアクセスされているとき、表示プロセッサ21からはアクセスできないためである。表示に使用されているラインメモリとは別のラインメモリを設けることにより、表示中であっても表示プロセッサ12がラインメモリへアクセスできるようにしている。

【0043】

ラインメモリ16aまたはラインメモリ16bへのデータ転送後、ステップA7にて、ベタスクリーンY方向の座標サイズy1と、次に表示するライン番号Lとを比較する。(L+1)の値がy1よりも小さいときは、ステップA8にてライン番号Lに相当するベタスクリーンデータ格納先頭アドレスaddrをベタスクリーンX方向の座標サイズx1だけ加算して次のライン番号に相当するベタスクリーン格納先頭アドレスaddrを取得する。同期用ウェイト(ステップA9)は、現在表示に使用されているラインメモリ16aまたはラインメモリ16bがまだ使用中であるかどうか、つまり次の水平表示の始まりまで待ってからラインメモリへの書き込みを行うことで、ラインメモリへの2重書き込みを制御している。上記に説明したラインメモリ16aまたはラインメモリ16bへの転送をy1回行うことで1画素分の表示を行うことが可能となる。

【0044】

次に複数のウィンドウ等を合成して表示する場合について説明する。

ベタスクリーンの表示では、メインメモリ12にある表示データをその先頭アドレスから順番に読み出して表示するだけであったが、表示プロセッサ21に与えるプログラムによっては、メインメモリ12の任意の位置のデータを任意の数だけ取り出したものを任意の組み合わせで表示することができる。例えばウィンドウシステムの場合、複数のウィンドウの表示データをそれぞれ別のアドレスに完成された形でメインメモリ12上に格納しておき、各ウィンドウの位置とプライオリティに従って、表示する際にリアルタイムに重ねあわせて表示することができる。

【0045】

ここでは図8のメモリマップに示すように、メインメモリ12上の任意のアドレス位置に背景データ、カーソルデータ、ウインドウ1データ、ウインドウ2偶数データ、ウインドウ2奇数データ等の各種表示データが完成された形で格納されているものとする。これらの表示データのうち、合成したときに表示されるデータのみを読み出してラインメモリに転送する。ウインドウ2偶数データ、ウインドウ2奇数データとは、NTSC信号のようなインタレース信号をメインメモリ12上に取り込むときに、フィールドごとに偶数データ、奇数データとして取り込んだ場合等のデータ構造である。ただし、カーソルの表示については後述する。

【0046】

図9は複数のウインドウを合成表示するフローチャートである。図8の各種データを合成したときに表示されるデータのみを読み出して1画面分表示する動作である。

ウインドウ座標やプライオリティ等の表示データはいつ変更されるかわからない任意のデータであるため、1画面を表示するごとに、ステップB1にて表示プロセッサ21によるメインメモリ12からデータメモリ20へのデータ転送を行う。また1画面を表示するために、ステップB2にて画面Y方向の座標サイズy1を取得し、ステップB3にて現在表示中の水平ライン番号の次の水平ライン番号をライン番号Lとして取得する。 α ブレンディングするか否かをB4にて判断し、 α ブレンディングしないのであれば通常ライン転送（ステップB5）を行い、 α ブレンディングするのであれば α ブレンディングライン転送（ステップB6）を行う。次にステップB7にて、表示するライン番号Lと画面Y方向の座標サイズy1を比較し、y1回のループが終了していなければラインメモリへの2重書き込みを制御する同期用ウェイト（ステップB8）の処理を行い、以上の処理をy1回行うことにより1画面分の表示を行う。

【0047】

図10は α ブレンディングなしの通常ライン転送のフローチャートである。図11（A）は α ブレンディングなしの表示画面例、（B）はライン番号Lのおけ

るラインメモリのメモリマップである。図11のライン番号Lを次に表示されるライン番号Lとして、そのライン番号上での通常ライン転送を以下に説明する。ステップC1において、表示プロセッサ21が、αブレンディングなしのライン番号L上の各表示データどうしの境界ポイントおよびポイント数を算出する。データメモリ20上に各ウィンドウの表示データが転送されて、右上座標、左下座標、X方向座標サイズ、Y方向座標サイズ、プライオリティ等から、境界ポイントやポイント数が算出される。あるいは、メインCPU11によって、あらかじめ算出しておいたデータをデータメモリ20に転送しておき、データメモリ20を参照するだけで取得してもよい。

【0048】

このときの境界ポイントを $xpt[]$ （ $[]$ 内は配列順を示す数を記入する）、境界ポイントの数を xpm として定義する。図11(A)に示すように、ライン番号L上の境界ポイントは $xpt[0] = xs0$, $xpt[1] = xs1$, $xpt[2] = (xe1 + 1)$, $xpt[3] = (xe2 + 1)$, $xpt[4] = (xe0 + 1)$ となり、境界ポイント数 xpm は5となる。ステップC2にて、境界カウンタ xp をクリアし、ステップC3にてラインL上の左境界ポイント xpl を取得し、ステップC4にて左境界ポイントの最も近い右境界ポイント xpr を取得する。この xpl , xpr 間の表示データを判別し、ステップC5にてライン番号Lに相当する表示データ格納先頭アドレス $addr$ を取得する。最初は $xpl = xpt[0] = xs0$ であり、 $xpr = xpt[1] = xs1$ であることから、この表示データは背景データであることが判別でき、 $addr = back_addr + xl * L + xs0$ によりライン番号Lに相当する背景データ格納先頭アドレス $addr$ が算出できる。

【0049】

ステップC6にてライン番号Lが偶数か奇数かを判断し、ラインメモリ16aへのデータ転送（ステップC7）またはラインメモリ16bへのデータ転送（ステップC8）の切り替えを行う。ラインメモリ16aおよびラインメモリ16bへのデータ転送サイズは、表示範囲が xpl , $xpr - 1$ であるため、 $xpr - xpl$ となる。ラインメモリ16aまたはラインメモリ16bへの書き込み位置

は x_{p1} であるので、ラインメモリ 16 a またはラインメモリ 16 b へのデータ転送は $addr$ から $(x_{s1} - x_{s0})$ のデータをラインメモリ 16 a またはラインメモリ 16 b の x_{s0} に転送することになる。右境界ポイント x_{pr} は次の x_{s1} , $(x_{e1} + 1)$ 間のデータを転送するときには左境界ポイント x_{p1} となるため、ステップ C9 にて $x_{p1} = x_{pr}$ とすることで左境界ポイント x_{p1} を取得することができる。既に説明した右境界ポイント x_{pr} の取得（ステップ C4）に移行し、これらの動作を x_{s1} , $(x_{e1} + 1)$, $(x_{e1} + 1)$, $(x_{e2} + 1)$, $(x_{e2} + 1)$, $(x_{e0} + 1)$ の境界間でも同様に行うことにより、ライン番号 L の 1 ラインのデータ転送を行うことができる。ステップ C10 にて境界カウンタ x_p と境界ポイント数 x_{pm} の比較を行い、境界カウンタ x_p が境界ポイント数 x_{pm} と同じか大きくなることにより次ラインの処理へと移行する。

【0050】

図 12 は α ブレンディングを含んだライン転送のフローチャートである。図 13 は α ブレンディングを含む表示画面例である。図 13 (A) は α ブレンディングの表示画面例、(B) はライン番号 L のおける通常ラインメモリと α ブレンディング用ラインメモリのメモリマップである。図 13 のライン番号 L を次に表示されるライン番号 L として、そのライン番号上での α ブレンディングライン転送を以下に説明する。ステップ D1 において、表示プロセッサ 21 は、 α ブレンディングのあるライン番号 L 上での各表示データどうしの境界ポイントおよびポイント数を算出する。図 11 の通常表示画面例よりも境界ポイント数が 1 つ増えている。この境界ポイントやポイント数はデータメモリ 20 上に転送されて得られた各表示データの右上座標、左下座標、X 方向座標サイズ、Y 方向座標サイズ、プライオリティ等により算出される。あるいは、メイン CPU 11 によってあらかじめ算出しておいたデータをデータメモリ 20 に転送しておき、データメモリ 20 を参照するだけで取得してもよい。

【0051】

ライン番号 L 上の境界ポイントは $x_{pt}[0] = x_{s0}$, $x_{pt}[1] = x_{s1}$, $x_{pt}[2] = x_{s2}$, $x_{pt}[3] = (x_{e1} + 1)$, $x_{pt}[4] =$

$x_{e2}+1$), $x_{pt}[5] = (x_{e0}+1)$ となり境界ポイント数 x_{pm} は 6 となる。 α ブレンディングのない境界は通常ライン転送と同じであるため、 α ブレンディングのある境界カウンタ x_p のときについて説明する。ステップ D14 によって取得されるライン L 上の左境界ポイント x_{pl} は $x_{pl} = x_{pr} = x_{pt}[2] = x_{s2}$ であり、D4 による右境界ポイント x_{pr} の取得により $x_{pr} = x_{pt}[3] = (x_{e1}+1)$ となる。この表示データのライン番号 L に相当するウィンドウ 1 データ格納先頭アドレス $addr$ は、 $addr = win1_addr + (x_{e1} - x_{s1} + 1) * (L - y_{s1}) + (x_{s2} - x_{s1})$ と算出される (ステップ D5)。ライン番号 L が偶数か奇数かをステップ D6 にて判断し、ラインメモリ 16a へのデータ転送 (ステップ D7) またはラインメモリ 16b へのデータ転送 (ステップ D8) の切り替えを行う。

【0052】

ラインメモリ 16a またはラインメモリ 16b へのデータ転送サイズは、表示範囲が x_{pl} , $x_{pr}-1$ であるため、 $x_{pr} - x_{pl}$ となる。ラインメモリ 16a またはラインメモリ 16b への書き込み位置は x_{pl} であるから、ラインメモリ 16a またはラインメモリ 16b へのデータ転送は $addr$ から $(x_{e1} + 1) - x_{s2}$ のデータをラインメモリ 16a またはラインメモリ 16b の x_{s2} に転送することになる。データ転送終了後、そのデータに対して α ブレンディングする別データがあるか否かをステップ D9 にて判断する。ここではウィンドウ 1 とウィンドウ 2 が α ブレンディングであり、この表示データのライン番号 L に相当するウィンドウ 2 偶数データ格納先頭アドレス $addr$ は、 $addr = win2e_addr + (x_{e2} - x_{s2} + 1) * (L - y_{s2})$ と算出される (ステップ D10)。

【0053】

ライン番号 L が偶数か奇数かをステップ D11 で判断してラインメモリ 16c へのデータ転送 (ステップ D12) またはラインメモリ 16d へのデータ転送 (ステップ D13) の切り替えを行う。このときのラインメモリ 16c またはラインメモリ 16d は α ブレンディング用ラインメモリである。ラインメモリ 16c またはラインメモリ 16d へのデータ転送サイズは、表示範囲が x_{pl} , x_{pr}

-1であるため、 $xpr - xpl$ となる。ラインメモリ16cまたはラインメモリ16dへの書き込み位置は xpl であるから、ラインメモリ16cまたはラインメモリ16dへのデータ転送は、 $addr$ から $(xe1 + 1) - xs2$ のデータをラインメモリ16cまたはラインメモリ16dの $xs2$ に転送することになる。通常ラインメモリには α ブレンディングしないデータを、 α ブレンディング用ラインメモリには α ブレンディングするデータを別々に持つことができ、ハードウェアの α ブレンディング処理により合成表示を行うことが可能となる。以降の処理であるステップD14、ステップD15は通常ライン転送と同様である。

【0054】

カーソルの表示は上記に記載した動作手順によっても表示することができるが、上記の1ライン分の表示データをラインメモリに転送した後に、カーソルの座標、カーソルX方向サイズ、カーソルY方向のサイズ、カーソルデータ格納先頭アドレス $curs_addr$ 等を与えて、最後に合成表示させることによっても実現できる。 α ブレンディングのデータ上に表示する場合は通常のラインメモリと α ブレンディング用のラインメモリの両方に書き込むことで、カーソルの表示が行える。この方法ではカーソルは常に最上位の優先順位となり、処理速度を早くすることができる。以上が表示プロセッサ21の基本動作の説明である。

【0055】

次に表示プロセッサ21が行う他の動作について説明する。

まず表示データの拡大・縮小・スキップの処理について説明する。図3に示したように、表示プロセッサ21は内部に転送用バッファメモリを2組持っている。メインメモリ12から読み込まれた表示データは、まず1組目の転送用バッファメモリ25a、25bに格納され、次にもう1組の転送用バッファメモリ26a、26bに格納されたのちに、表示用のラインメモリ16へ格納される。この転送用バッファメモリ間の読み出しおよび書き込みは、表示プロセッサ21に与えるプログラムによって細かく制御できる。

【0056】

具体的には、1組目の転送用バッファメモリ25a、25b（読み出しメモリ

と呼ぶ)の読み出しカウンタのスタート/ストップ、もう1組の転送用バッファメモリ26a, 26b(書き込みメモリと呼ぶ)への書き込みカウンタのスタート/ストップ及び書き込みのする/しないを画素単位で任意の位置で行うことができる。これにより表示画像の拡大、縮小や、ある位置より右側の画像が右方向にずれて画像に穴が開いたように見える表現(スキップと呼ぶ)およびそれらを混在させた表示データに変化させることができる。

【0057】

拡大・縮小・スキップの動作は図14に示すコントロールデータの動作説明図に示すように制御される。コントロールデータは1画素につき2ビットの情報を持ち、画素単位で転送用バッファメモリ25a, 25b, 26a, 26b間の読み出しカウンタおよび書き込みカウンタおよび書き込みのする/しないを制御する。図15は拡大・縮小・スキップを行わない等倍のときの転送用バッファメモリ間の転送動作であり、この場合、コントロールデータとして“00”を与え続ける。すると、読み出しカウンタ・書き込みカウンタとも1ずつカウントアップされていき、読み出しメモリと同じデータが書き込みメモリに書き込まれて等倍の転送となる。

【0058】

縮小を行う場合は、コントロールデータの省きたい画素に対応するデータを“01”にする。縮小動作を示す図16において、書き込みメモリには、0、1、2、3までは順に表示データが書き込まれるが、3の位置のコントロールデータが“01”であるため、書き込みカウンタがストップし、次に3の位置に4を重ね書きする。これで表示データが1画素分だけ縮小する。コントロールデータに1画素おきに“01”を設定すれば画像の水平方向は1/2に縮小するし、部分的に“01”を設定する割合を変えれば、例えば画像が円柱形になったりする。

【0059】

拡大を行う場合には、コントロールデータの対応する位置に“10”を設定する。図17において、書き込みメモリには、0、1、2、3までは順に表示データが書き込まれるが、3の位置のコントロールデータが“10”であるため、読み出しカウンタがストップし、次に3の隣にもう一度3が書かれる。これで1画

素分の拡大が行われる。

【0060】

コントロールデータが“11”のときはスキップである。図18で0、1、2まではそのまま書かれるが、3の位置のコントロールデータが“11”であるため、読み出しアドレスが停止する。このため3の表示データは右隣の画素に書かれることになる。さらに書き込みメモリへの書き込みが行われず、書き込みメモリの3の位置には何も書かれない。これで1画素分のスキップが行われる。

【0061】

以上のようにコントロールデータの値を設定することにより、拡大・縮小・スキップが可能であり、また図19～図21のように拡大・縮小・スキップを混在して設定することにより一部は拡大するが他の一部は縮小するといったような複雑な表示データの変形が行える。

【0062】

ところで拡大・縮小率が水平方向に一定であることは多いが、この場合コントロールデータは同じパターンの繰り返しとなる。本実施形態では繰り返すパターンと繰り返しポイントを設定することにより、1水平ライン分のコントロールデータを書き込むのに比べ、少ないデータで拡大・縮小等の指定をすることができる。例えば0.75倍に縮小する場合には図22のようにコントロールデータは“00”、“00”、“00”、“01”の繰り返しとなる。この場合、この4画素分のコントロールデータと4画素単位で繰り返しが行われるように繰り返しポイントの設定をすることにより、同じコントロールデータが繰り返し使用され、縮小動作が行われる。同様に図23は1.75倍に拡大する場合である。

【0063】

次に、本実施形態ではビデオ入力を2系統持っているが、表示プロセッサ21は、これよりビデオ映像データの取り込みを行うことができる。ビデオ映像信号はA/D変換された後、ビデオ入力用ラインメモリに格納される。ビデオ入力用ラインメモリはビデオ入力1系統につき2本あり、他のラインメモリと同様に読み出しと書き込みを行うメモリを交互に切り替えて使用する。ビデオ入力用ラインメモリに書き込まれたビデオデータは、表示用プロセッサ21によって読み出

され、表示プロセッサ21内で拡大・縮小・スキップ処理などを行った後、ラインメモリ16に転送される。

【0064】

次にデータ処理回路13について説明する。メインメモリ12に格納されている表示データは通常のRGB形式のデータだけでなく、さまざまなデータ形式で格納されている。表示プロセッサ21によってメインメモリ12から表示データが読み出されラインメモリ16に書き込まれる間に、YUVデコーダ27a、ランゲンス展開回路27b、カラー伸長回路27c、カラーパレット27d、27eの処理回路があり、そこで各種データ形式の表示データはRGB形式に変換されてラインメモリ16に格納される。どのデータ処理回路によって変換を行うかは、表示プロセッサ21が画素単位にセレクタ28に指示して選択させる。カラーパレットは複数持つことができ、例えばウィンドウごとに使うパレットを変えることができる。また、さらに他のデータ処理回路を追加することで、さまざまな表示データのフォーマットに対応することができる。

【0065】

データ処理回路13を通過した表示データはラインメモリ16に書き込まれるが、表示データのうちのいくつかの値を、実際には表示されないライトスルーデータとして設定することができる。メインメモリ12やデータバッファ15から表示プロセッサ21が表示データをラインメモリ16に転送する際、ライトスルーデータがあると、その画素についてはラインメモリ16への書き込みを行わない。これは矩形でない画像、例えばマウスカーソルなどの表示に有効である。

【0066】

次に使用ライン情報を用いて画面表示を行う動作について説明する。通常、表示用のラインメモリは2本一組で動作する。これは表示のために読み出しを行っているラインメモリに対して、表示プロセッサ21が書き込みアクセスを行うことができないため、読み出しを行っているラインメモリとは別のもう一方のラインメモリに次のラインの表示データの書き込みを行う。表示するラインが変わるたびに、この読み込みと書き込みを行うラインメモリを交互に入れ替えて表示を進めていく。ところが図4(B)、(C)のように複数の画面を合成して表示

する場合で、特に背景を表示しないときには、ラインメモリへの表示データの書き込みは、ウインドウを表示する部分についてのみ行われ、その他の部分には前のラインの表示データが残ったままになることがある。そのため書き込み前にラインメモリのクリアが必要となり、そのための時間が必要になってくる。使用ライン情報は、このラインメモリのクリア作業を不要にするものである。

【0067】

使用ライン情報は、ラインメモリ上の各画素の表示データに1対1で対応し、その表示データが何ライン目の表示で使用されるデータであるかを表す情報である。表示データ1画素に対応する使用ライン情報は（画面の垂直方向の画素数+1）を表現できるビット数（画面サイズが1280×1024ならば11ビット）以上で、それが各ラインメモリに表示データと同じ画素数分、つまり水平画素数分だけである。

【0068】

図24は、使用ライン情報を格納する表示メモリ部14を示すブロック図である。ラインメモリ16a～16fにはそれぞれコンパレータ31～36、AND回路37～42が接続されている。ラインメモリ16e、16fは後述する背景データを格納するメモリである。コンパレータ31～36は、表示ライン数と使用ライン情報を比較し、値が一致する場合は論理値1を出力し、不一致の場合は論理値0を出力する。AND回路37～42は、論理値1が入力されると、表示データをそのまま出力し、論理値0が入力されると表示データを出力しない。

【0069】

以下、画面表示動作について図25をもとに説明する。図25（A）は表示画面例、（B）は使用ライン情報がNの場合のラインメモリのメモリマップと出力データ、（C）は使用ライン情報がN+2の場合のラインメモリのメモリマップと出力データ、（D）は使用ライン情報がN+4の場合のラインメモリのメモリマップと出力データである。同図（B）に示すように、（N-1）ライン目の表示を行っている間、表示プロセッサ21はラインメモリにNライン目の表示データを書き込む。Nライン目にはウインドウ1があり、ウインドウ1の表示データ6書き込む際に同時に使用ライン情報にNを書き込む。Nライン目を表示する際

には、ラインメモリの1画素毎に、表示中のライン番号Nと使用ライン情報との比較を行い、それが等しい場合のみ表示データ有効とみなし、ラインメモリ中の表示データを出力する。

【0070】

次に同じラインメモリに書き込みを行うのは、ラインメモリを2本交互に使用するため(N+2)ライン目である。同図(C)に示すように、(N+2)ライン目には、ウインドウ1とウインドウ2の2つがあり、その表示データと使用ライン情報には(N+2)を書き込む。これを同様にして表示を行う。

【0071】

次に(N+4)ライン目の書き込みを行う。(N+4)ライン目はウインドウ2についてのみであり、同図(D)に示すように、表示データと使用ライン情報に(N+4)の書き込みを行う。このとき(N+2)ライン目で書き込んだウインドウ1のデータが残ったままになっており、何らかの工夫を行わない場合はこれが表示されてしまい、間違った表示となる。ところが本実施形態では、この古いウインドウ1の部分の使用ライン情報は(N+2)のままなので無視されて、ウインドウ2のみ正しく表示される。

【0072】

このようにしてすべてのラインについて表示を行うが、垂直帰線期間毎にすべてのラインメモリの使用ライン情報をクリアする必要がある。これは前の垂直表示期間の表示データが表示されるのを防ぐためである。なおクリアは、使用ライン情報として使用されていない値を書き込むことによって行う。

【0073】

次に同じパターンの繰り返し表示について説明する。ウインドウシステムの背景画面などによく見られるが、水平方向に同じパターンが繰り返し表示されることがよくある。この場合ラインメモリ16から読み出す読みだしアドレスを任意の範囲でループできる様にする事で、特定のパターンを繰り返し表示できる。それにより特に背景データをメインメモリ12に格納している場合などは、読み出して来るデータ量を削減でき、メインCPU11のデータバスのトラフィックを低下させることができる。この機能を使用する際には、通常のラインメモリの

他に繰り返しパターンを格納する専用のラインメモリ 16 e, 16 f が 2 本一組必要である。よってラインメモリは最低で 4 本、 α ブレンディングを同時に使用する場合は最低 6 本必要になる。この特定パターンの繰り返し表示機能について以下に説明する。

【0074】

図 26 は、背景を繰り返し利用する場合の動作説明図である。N ライン目についてラインメモリの書き込みを行う場合、まず通常と同様にウインドウデータを収納するラインメモリにウインドウの表示データおよび使用ライン情報 N を書き込む。次に背景データを収納するラインメモリに背景の表示データと使用ライン情報 N を書き込み、さらに繰り返しポイントを設定する。繰り返しポイントの設定方法にはいくつかの方法が考えられ、専用のレジスタを設けたり、使用ライン情報や表示データに通常と区別できる値を書き込んだり、専用のラインメモリを用意するといった方法が考えられる。

【0075】

表示の際には、まずウインドウデータを収納するラインメモリの使用ライン情報を表示中のライン番号と比較する。一致すればウインドウの表示データを出力し、一致しない場合は背景データを出力する。背景データは図示していないが内部の背景データ読み出しカウンタによって示される背景データが出力される。この読み出しカウンタの値が繰り返しポイントの値と一致したならば、読み出しカウンタの値がクリアされる。これにより出力される背景データは、背景データを収納するラインメモリの最初に戻り、背景データがこの範囲で繰り返して出力される。

【0076】

次にデータバッファ 15 について説明する。表示データはメインメモリ 12 上に格納するのが通常であるが、カーソルなどサイズが小さくパターンが決まっている表示データはデータバッファ 15 に格納するとよい。データバッファ 15 に格納された表示データは、表示プロセッサ 21 によってラインメモリ 16 に書き込むことができる。またラインメモリ 16 ではなく表示プロセッサ 21 のプログラムメモリ 19 やデータメモリ 20 やメインメモリ 12 に転送することもできる

ので、カーソルの表示など限らず汎用に使用することができる。

【0077】

また、 α ブレンディングで2画面の混合比率を設定する方法にはいくつか考えられる。ひとつは混合比率を格納する専用レジスタを用意して、 α ブレンディングするときそのレジスタより混合比率を読み出す方法がある。その場合、混合比率が変化するたびに表示プロセッサ21がレジスタの内容を書き換える必要がある。他には混合比率を複数格納するLUTを用意し、ラインメモリに表示データを書き込む際に、そのLUTの呼び出しアドレスと一緒に画素単位でラインメモリに書き込む方法、または直接混合比率をラインメモリに画素単位で書き込んでしまう方法などが考えられる。

【0078】

【発明の効果】

請求項1の発明によれば、表示をする際に必要な部分の表示データをメインメモリ内から取り出して使用するため、メインメモリ内の任意の位置のデータを取り出して組み合わせることが可能である。この制御はすべて表示制御部が行うので、主制御部が表示のためにスクリーン上に複数のウィンドウを同時表示する等の際のソフトウェアにおける処理負荷を低減でき、各ウィンドウの移動や切り替えを高速化できる。

【0079】

請求項2の発明によれば、ラインメモリ上のデータを読み出す際、それがライン方向に対して繰り返すようなデータであった場合（ウィンドウシステムにおける背景等）、読み出しラインメモリアドレスを任意の位置でループできるため、冗長な処理が不要になり、処理の高速化が図れる。

【0080】

請求項3の発明によれば、カーソルや繰り返し背景などをデータバッファメモリに収納しておけるため、決まりきったデータをメインメモリから読み出す必要が無い場合、データバスの負荷を減らし、冗長な処理が不要になり、処理の高速化が図れる。

【0081】

請求項4の発明によれば、表示データを読み出す際に拡大縮小処理をするため、表示用データに対する拡大縮小処理を事前にする必要が無く、バスの使用効率を上げられる。また、ビデオ入力映像を表示する場合に映像サイズの変更が必要となるのが常であるが、出力段に拡大縮小処理を掛けることで拡大縮小回路がより有効に利用できる。また、この事によりビデオデータを常にフルサイズで取り込みながら、そのデータを一旦フレームメモリなどに転送することなく表示は任意のサイズが行える。

【0082】

請求項5の発明によれば、第1バッファメモリからの読みだしアドレスカウンタを所定の順に停止／動作を繰り返すことにより、一定倍率の拡大・縮小が簡単な処理で行うことができ、処理の高速化ができる。

【0083】

請求項6の発明によれば、表示制御部は、格納情報のデータ形式情報に基づいてデータ変換ができるので、表示用データを収納する形式などに制限が無いため、データメモリ上に収納されている表示キャラクタ等をわざわざフレームバッファ等に転送する必要が無く、処理の高速化が図れる。

【0084】

請求項7の発明によれば、前記表示制御部に必要なプログラムとデータを格納するプログラムメモリとデータメモリとを備えるので、メインメモリから処理の度にメインメモリからデータを読み出す必要がなく、データバスの使用回数を減らし、処理の高速化を図ることができる。

【0085】

請求項8の発明によれば、前記表示制御部は、前記プログラムメモリとデータメモリに必要な情報をメインメモリから転送させるため、画面モードあるいはグラフィック領域の変更に柔軟に対応できる。容量を越えたプログラムあるいはデータは、メインメモリから読み出せばよいので、小容量で済み、コンパクトあるいは低コストでシステムを構築できる。

【0086】

請求項9の発明によれば、各ラインメモリに表示データを転送する際、そのデータを使用するライン番号を同時に、1ドット毎に対応した使用ライン情報メモリに書き込み、表示する際に表示しようとしているラインの番号と比較することによりラインメモリ上のデータが有効であるかどうかを判別することで、ラインメモリを使用する前に毎回ラインメモリの内容をクリアする必要が無くなり、処理の高速化が図れる。各ラインの表示毎にラインメモリ内のデータを消去する必要がなく、垂直帰線期間毎にすべてのラインメモリの使用ライン情報を消去するだけでよいので処理の高速化を図れる。

【図面の簡単な説明】

【図1】

本発明に係る画面表示装置の一実施形態を示すブロック図である。

【図2】

この画面表示装置のデータ処理回路と表示メモリ部を示すブロック図である。

【図3】

この画面表示装置の表示プロセッサを示すブロック図である。

【図4】

(A)～(C)は、メインメモリの表示データとディスプレイの表示出力を示す説明図である。

【図5】

ベタスクリーンデータを1画面分表示するフローチャートである。

【図6】

ベタスクリーンの表示画面例である。

【図7】

ベタスクリーンデータが格納されているメインメモリのメモリマップである。

【図8】

各種表示データが格納されているメインメモリのメモリマップである。

【図9】

複数のウィンドウを合成表示するフローチャートである。

【図10】

α ブレンディングなしの通常ライン転送のフローチャートである。

【図11】

(A)は α ブレンディングなしの表示画面例、(B)はライン番号Lにおけるラインメモリのメモリマップである。

【図12】

α ブレンディングを含んだライン転送のフローチャートである。

【図13】

(A)は α ブレンディングの表示画面例、(B)はライン番号Lにおける通常ラインメモリと α ブレンディング用ラインメモリのメモリマップである。

【図14】

コントロールデータの動作内容を示す説明図である。

【図15】

拡大・縮小・スキップを行わない等倍のときの転送用バッファメモリ間の転送動作の説明図である。

【図16】

転送用バッファメモリの縮小動作を示す説明図である。

【図17】

転送用バッファメモリの拡大動作を示す説明図である。

【図18】

転送用バッファメモリのスキップ動作を示す説明図である。

【図19】

拡大・縮小・スキップの混在した転送用バッファメモリの動作を示す説明図である。

【図20】

拡大・縮小・スキップの混在した転送用バッファメモリの他の動作を示す説明図である。

【図21】

拡大・縮小・スキップの混在した転送用バッファメモリの更に他の動作を示す

説明図である。

【図22】

転送用バッファメモリの一定倍率の縮小動作を示す説明図である。

【図23】

転送用バッファメモリの一定倍率の拡大動作を示す説明図である。

【図24】

使用ライン情報を格納する表示メモリ部を示すブロック図である。

【図25】

(A)は表示画面例、(B)は使用ライン情報がNの場合のラインメモリのメモリマップと出力データ、(C)は使用ライン情報がN+2の場合のラインメモリのメモリマップと出力データ、(D)は使用ライン情報がN+4の場合のラインメモリのメモリマップと出力データである。

【図26】

背景を繰り返し利用する場合の動作説明図である。

【図27】

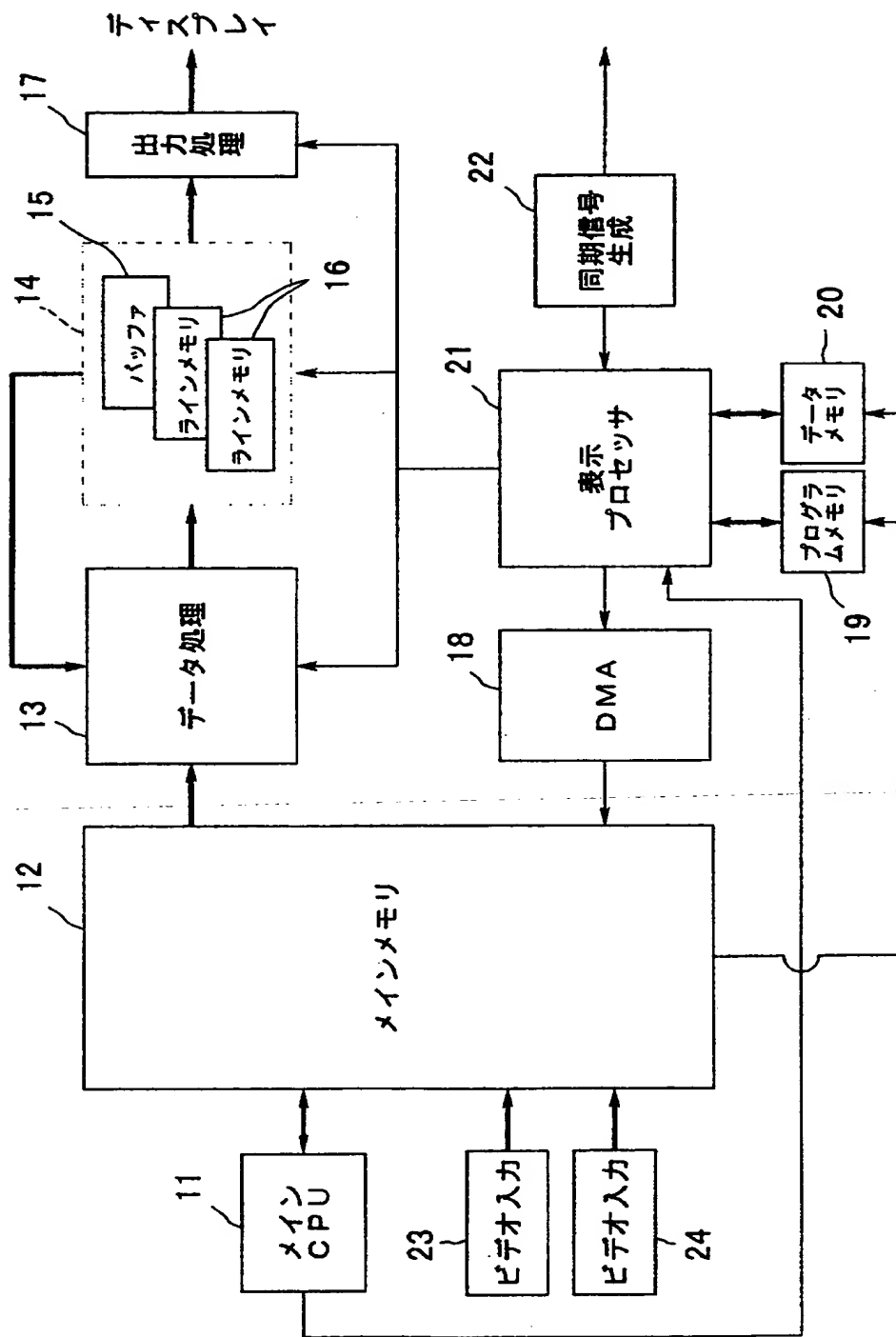
従来の画像表示装置の一例を示すブロック図である。

【符号の説明】

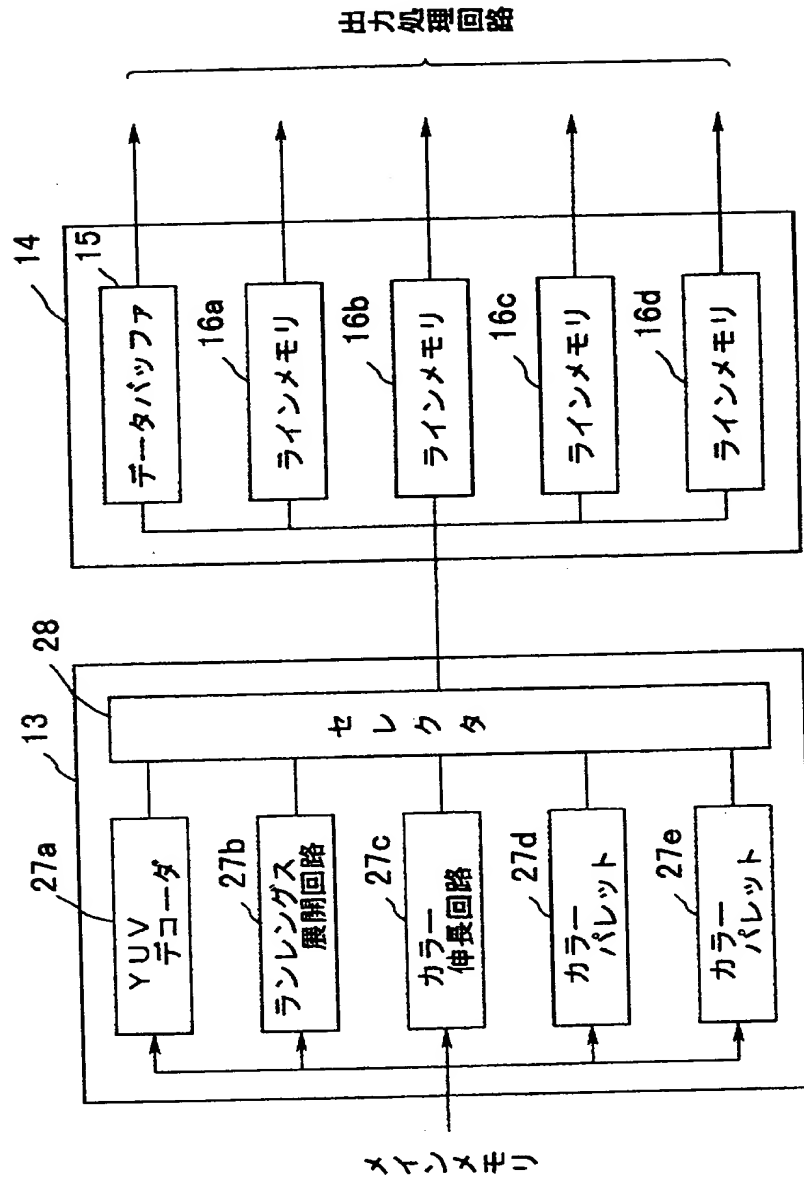
- 11 メインCPU
- 12 メインメモリ
- 13 データ処理回路
- 14 表示メモリ部
- 15 データバッファ
- 16 ラインメモリ
- 17 出力処理回路
- 18 DMA
- 19 プログラムメモリ
- 20 データメモリ
- 21 表示プロセッサ
- 22 同期信号生成回路

【書類名】 図面

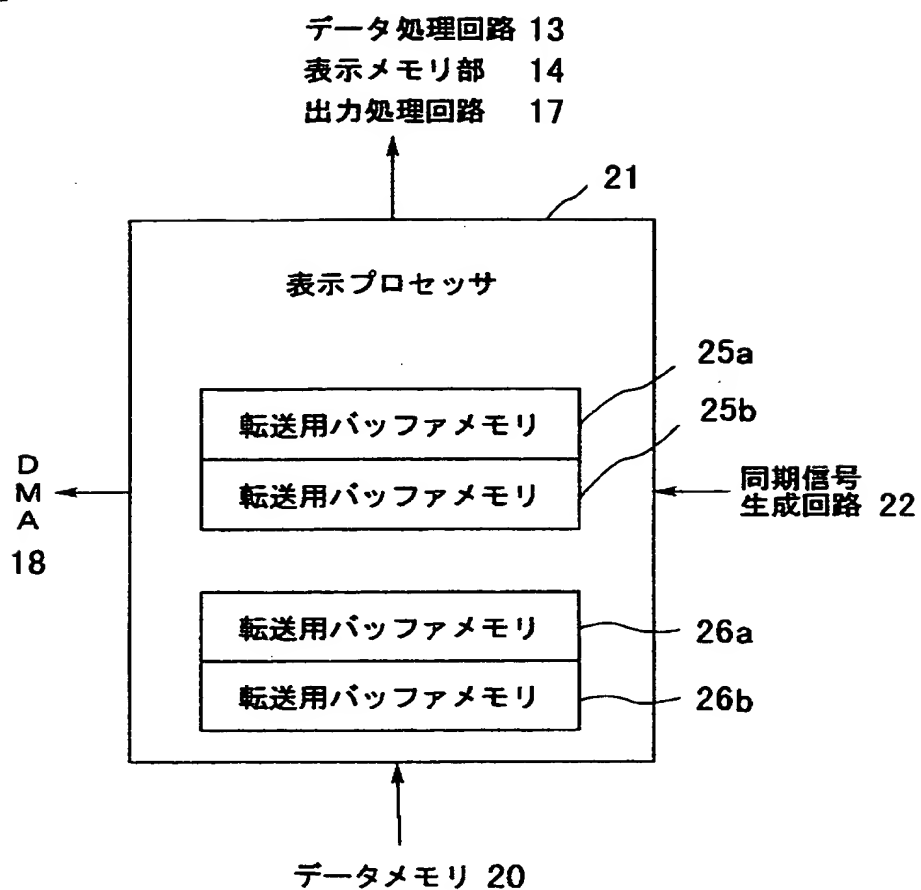
【図1】



【図2】

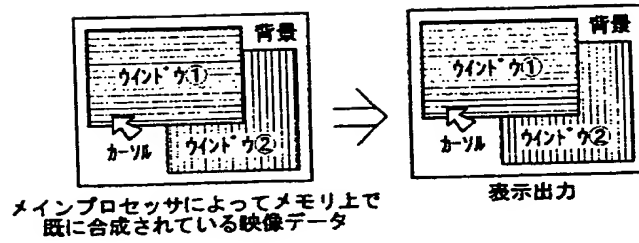


【図3】

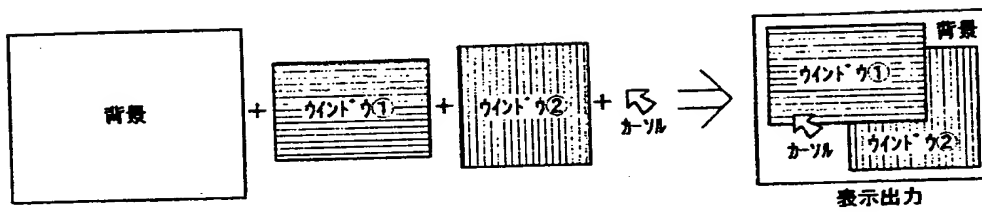


【図 4】

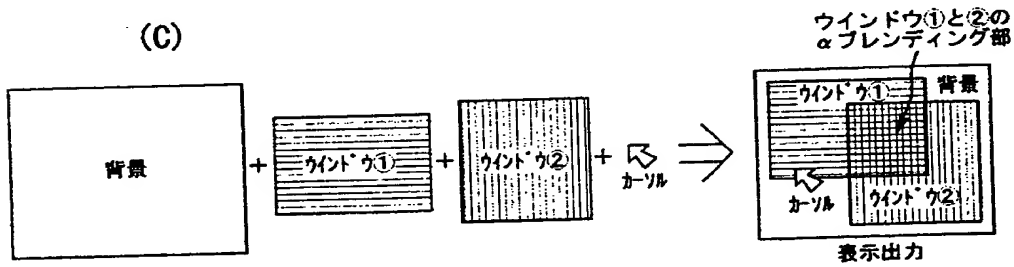
(A)



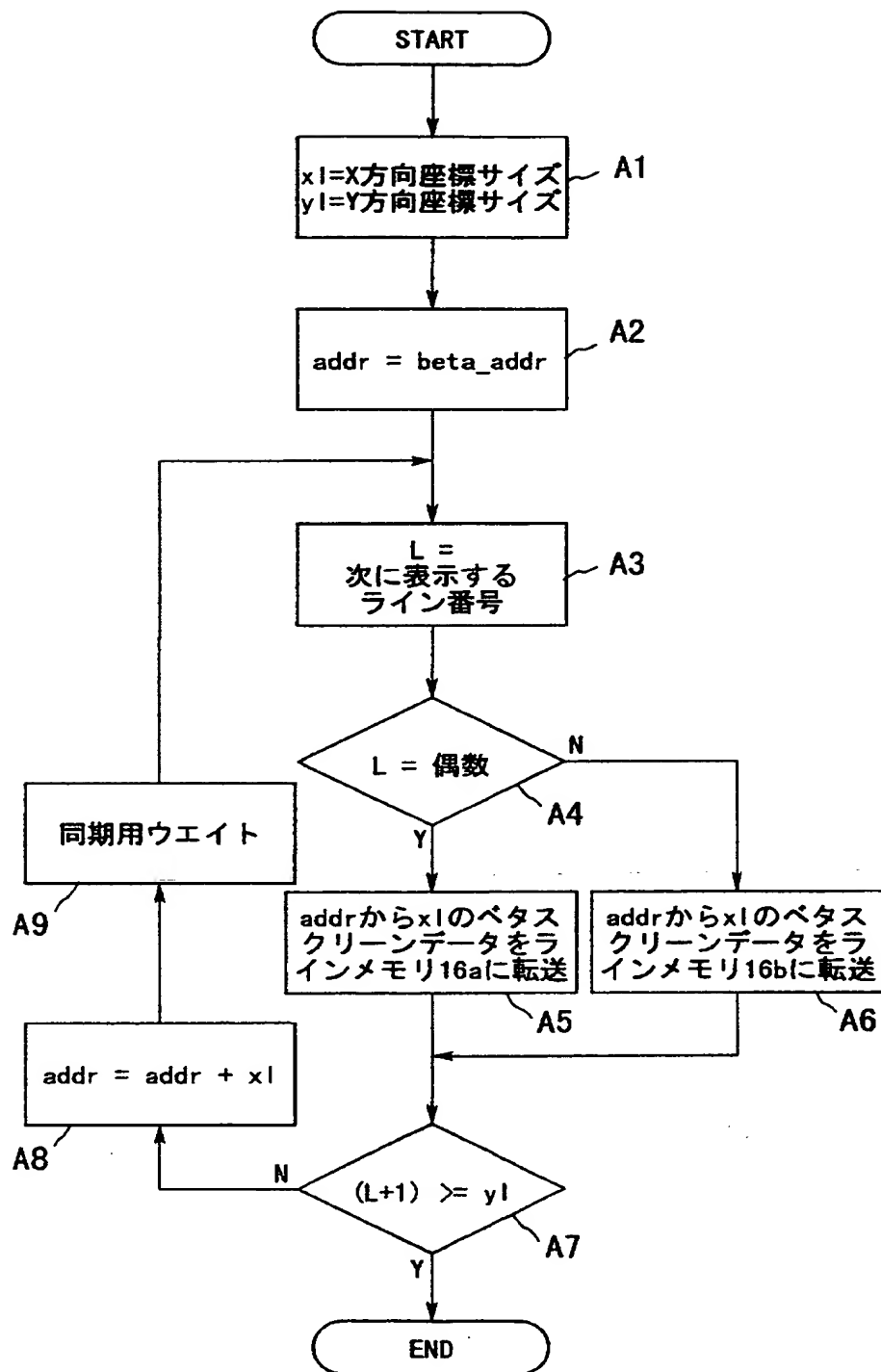
(B)



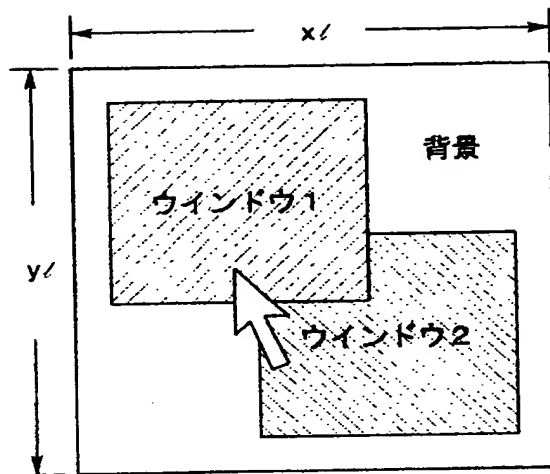
(C)



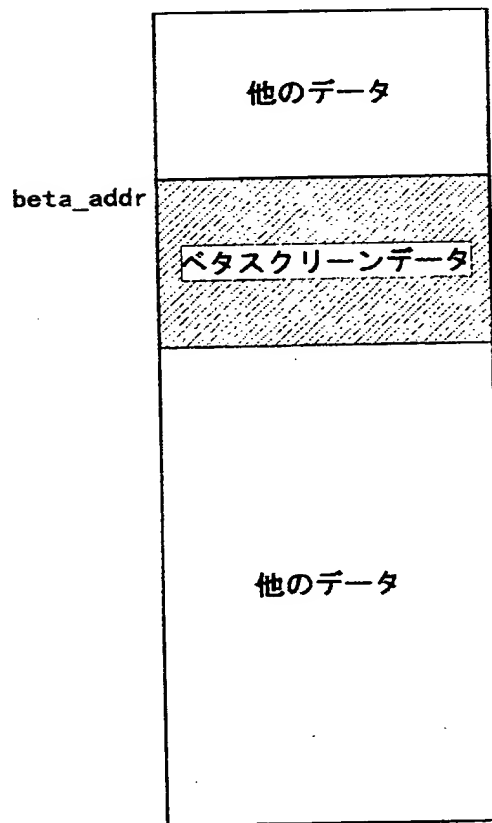
【図5】



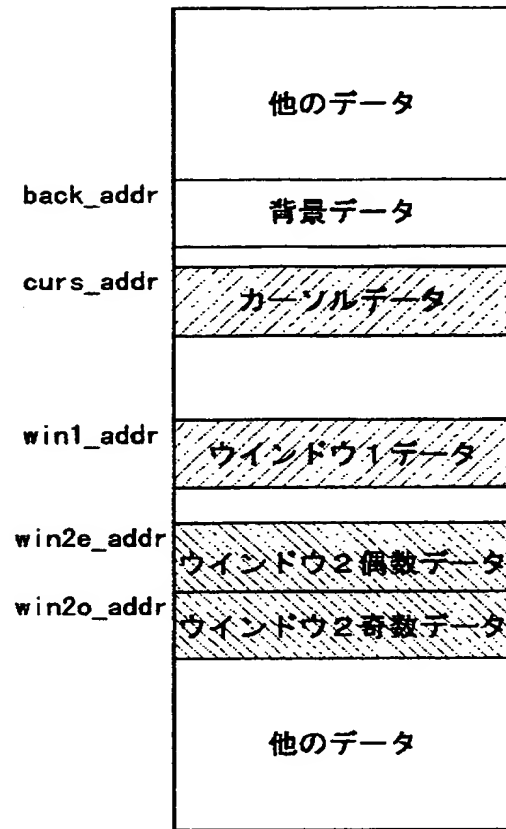
【図6】



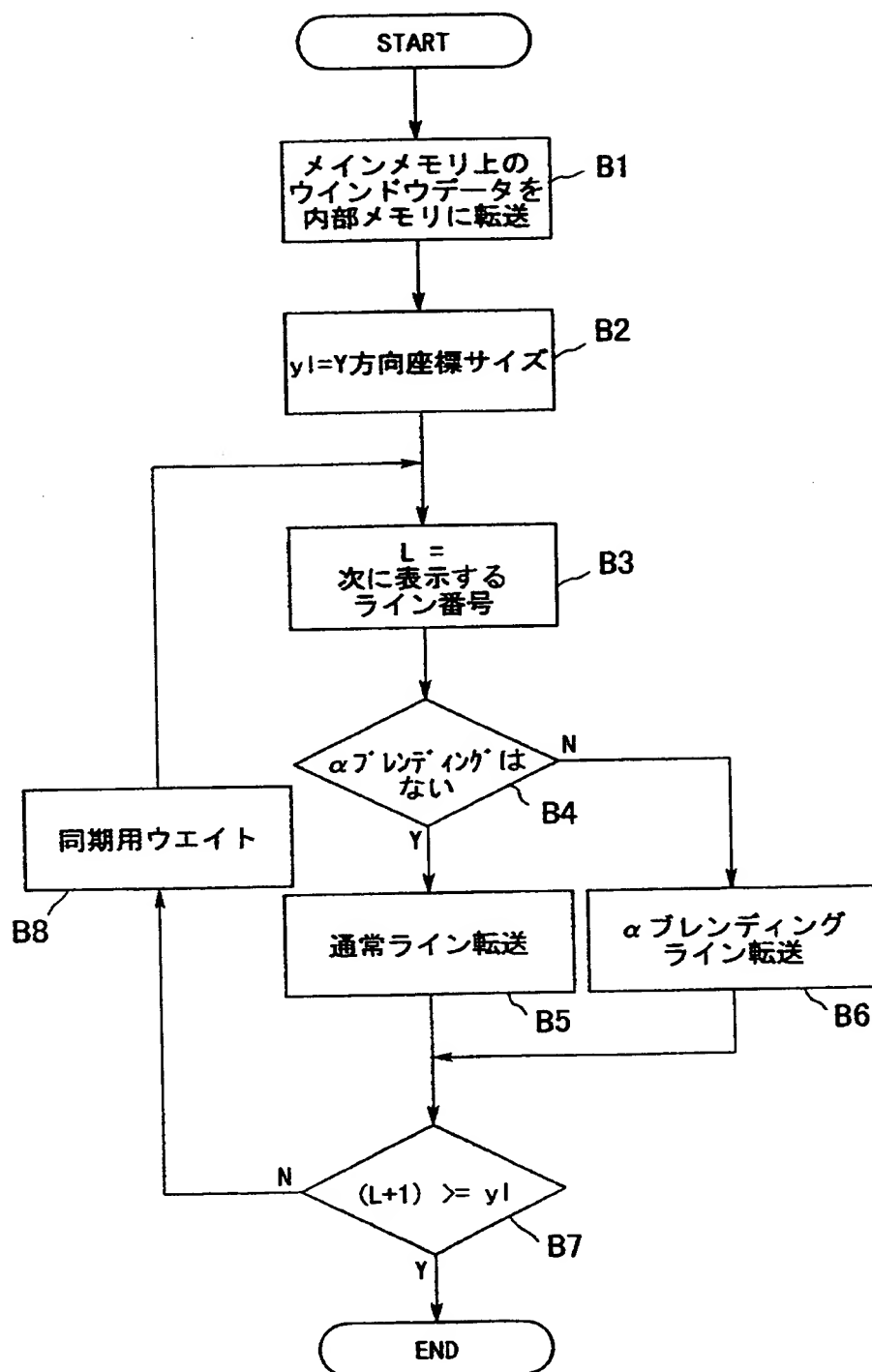
【図7】



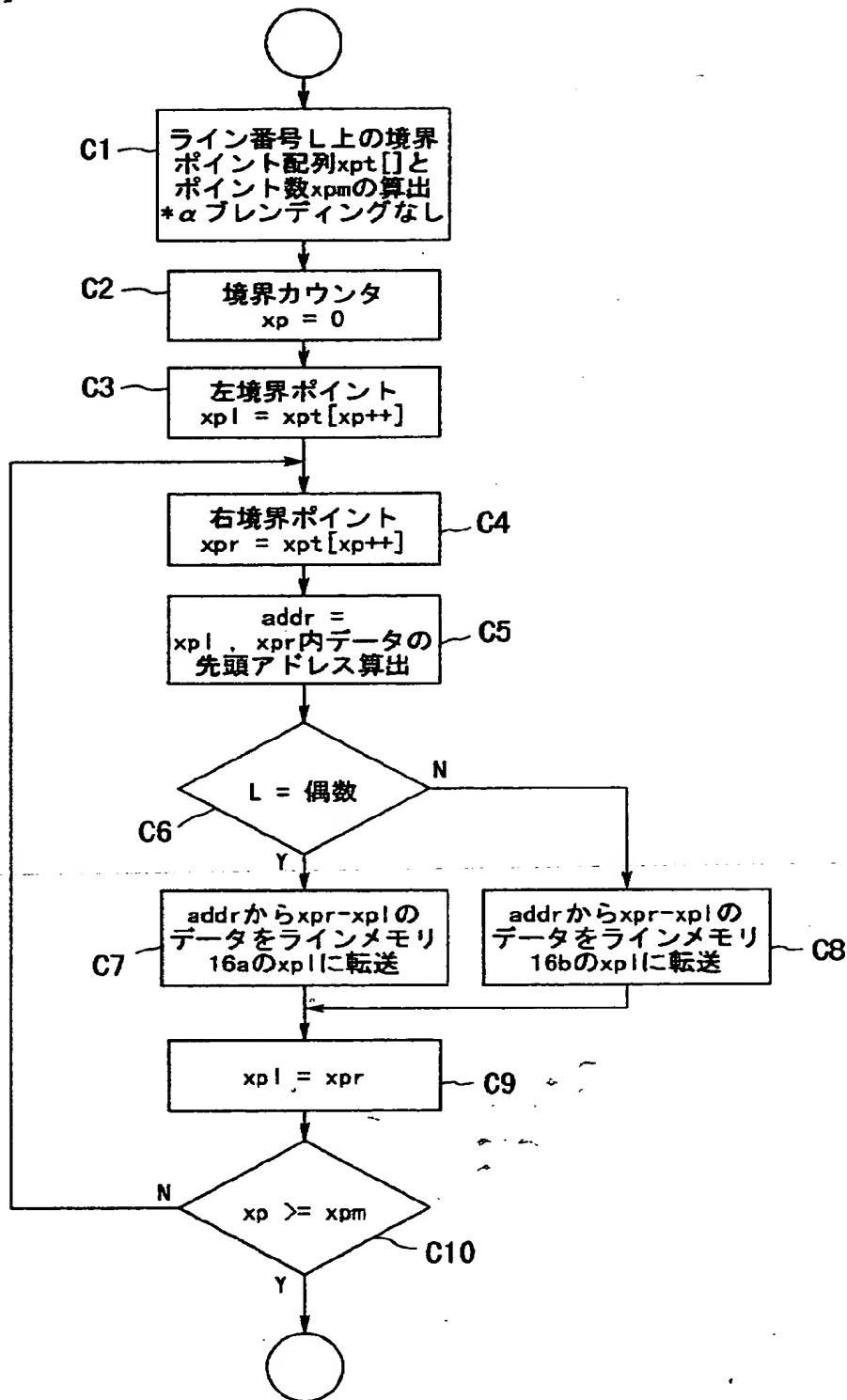
【図8】



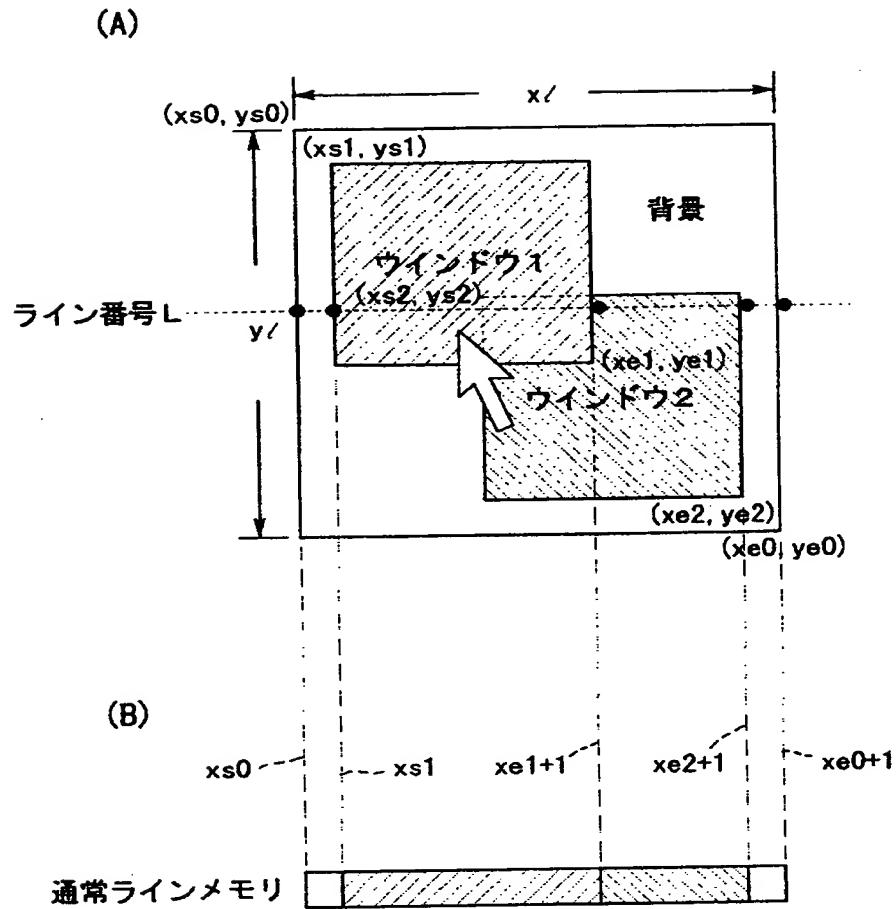
【図9】



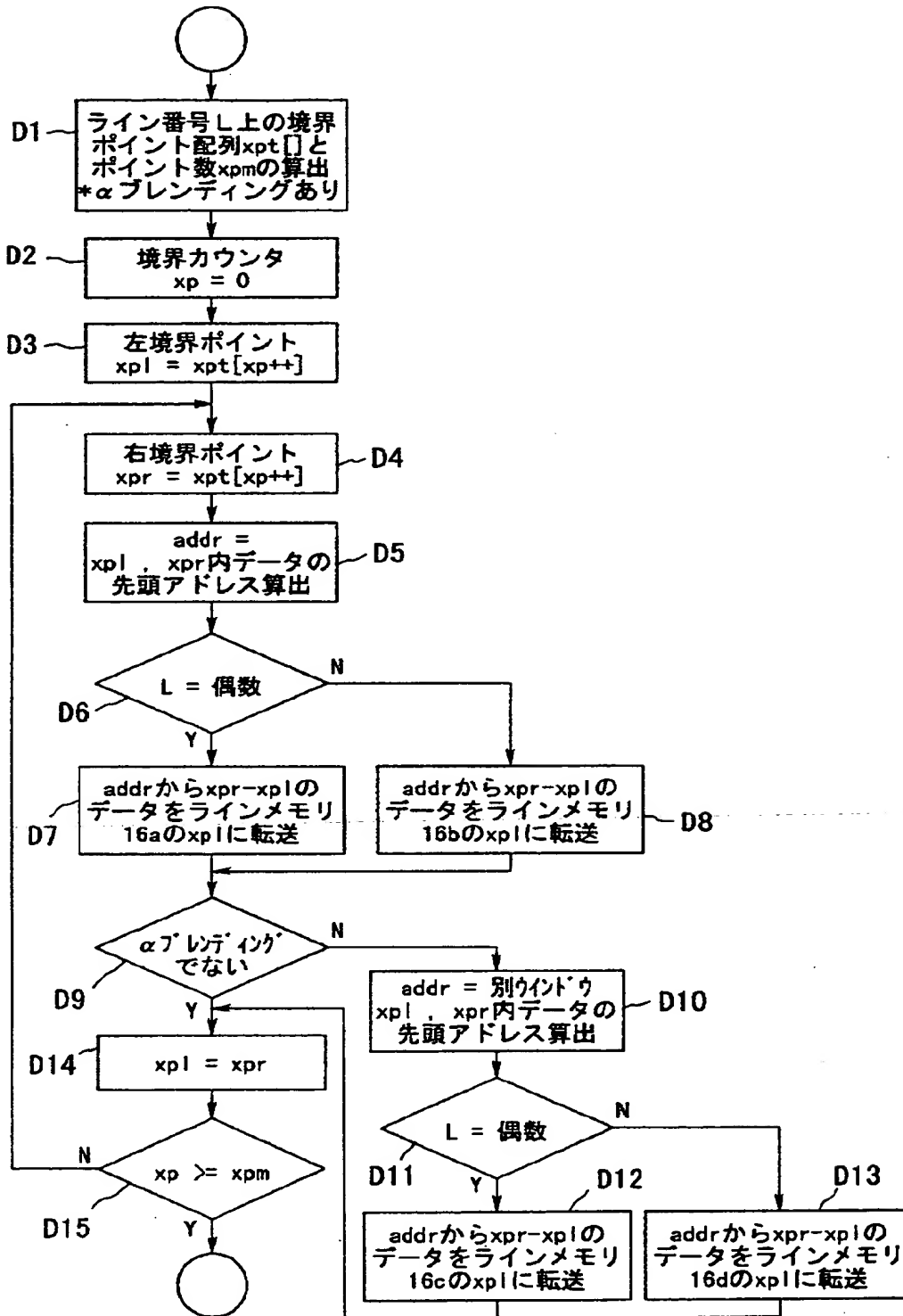
【図10】



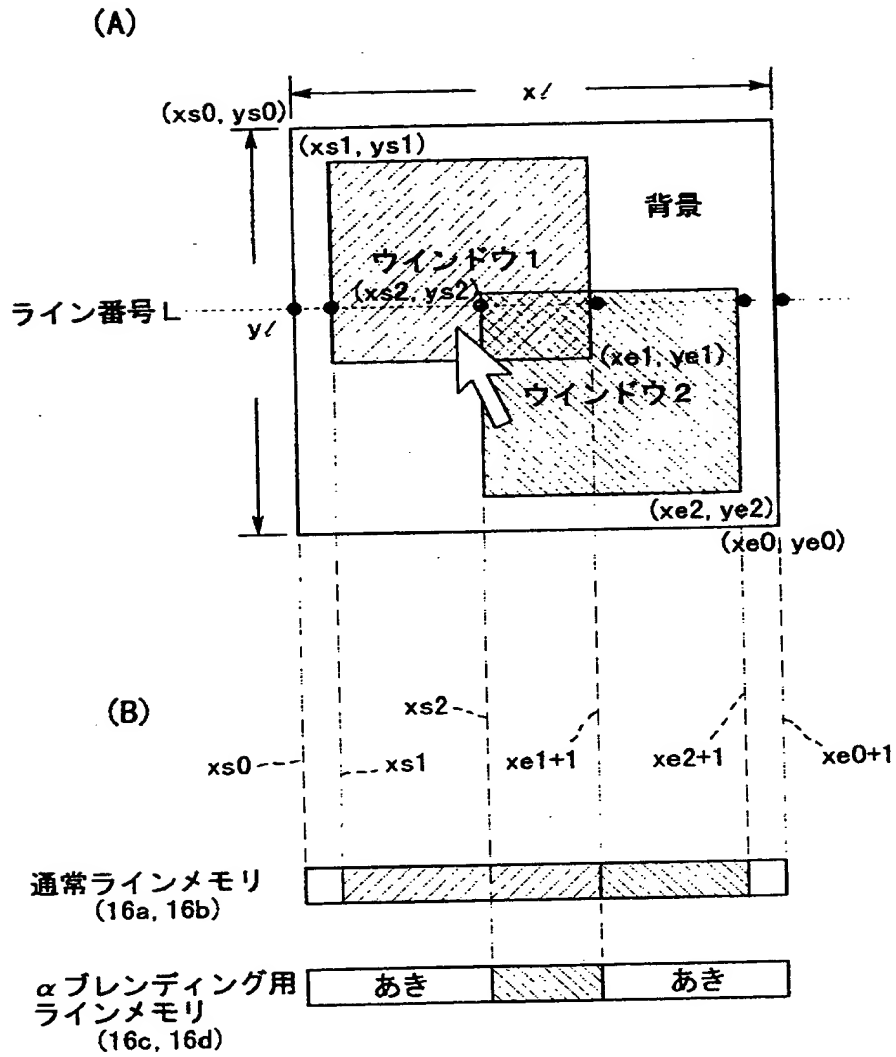
【図11】



【図12】



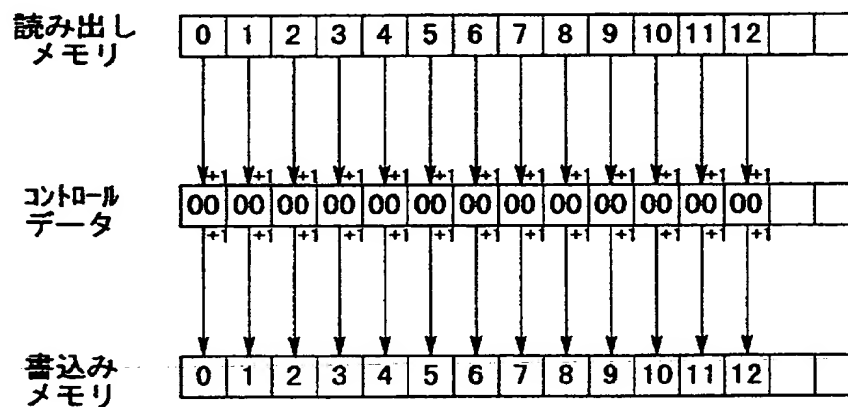
【図13】



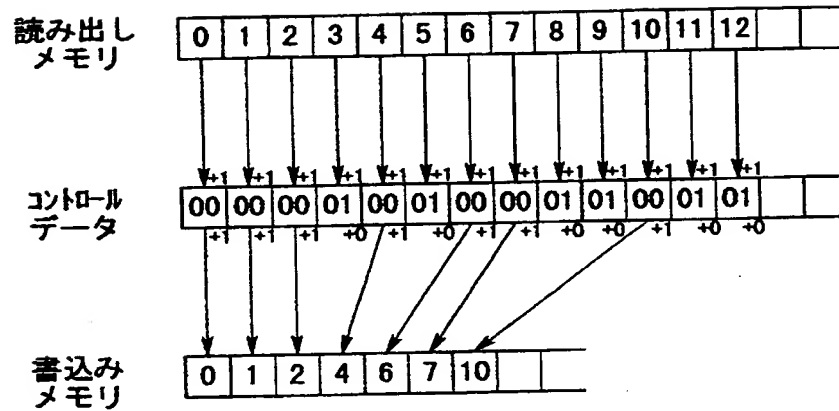
【図14】

| コントロール データ | 動作状態 | 処理後の各カウンタの動作 | | 書込み動作 |
|---------------|------|--------------|---------|-------|
| | | 読出しカウンタ | 書込みカウンタ | |
| 00 | ノーマル | +1 | +1 | する |
| 01 | 縮小 | +1 | +0 | しない |
| 10 | 拡大 | +0 | +1 | する |
| 11 | スキップ | +0 | +1 | しない |

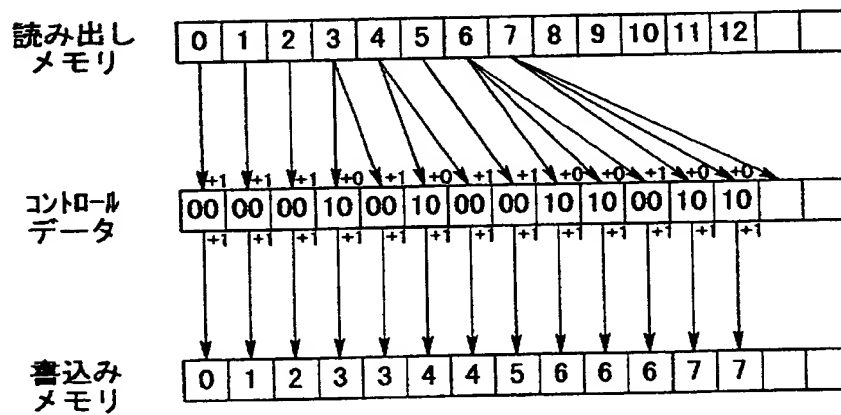
【図15】



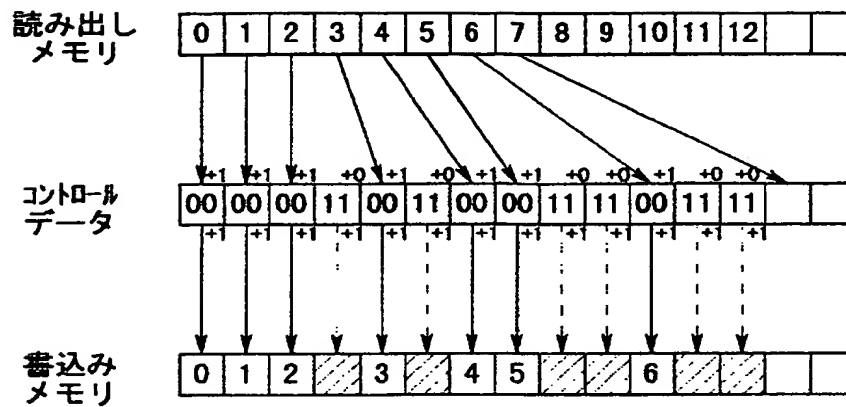
【図16】



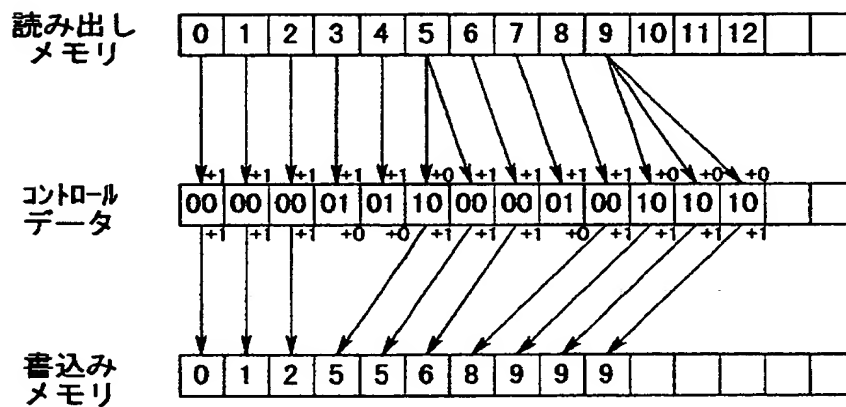
【図17】



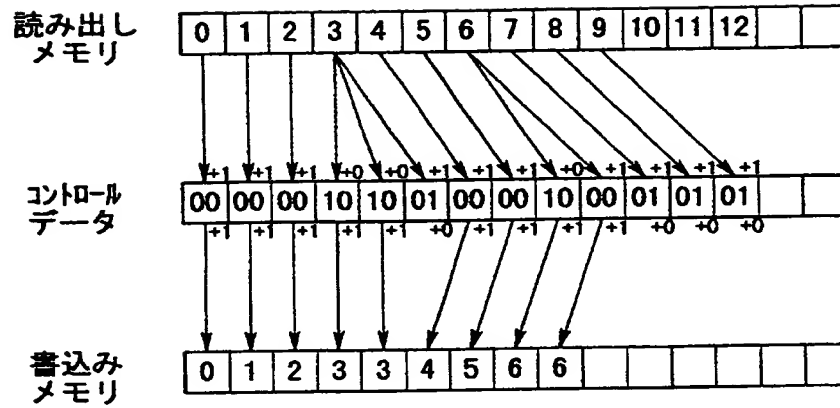
【図18】



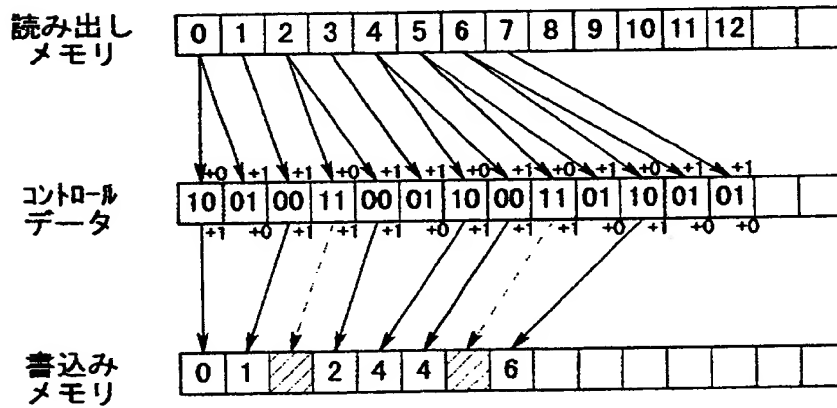
【図19】



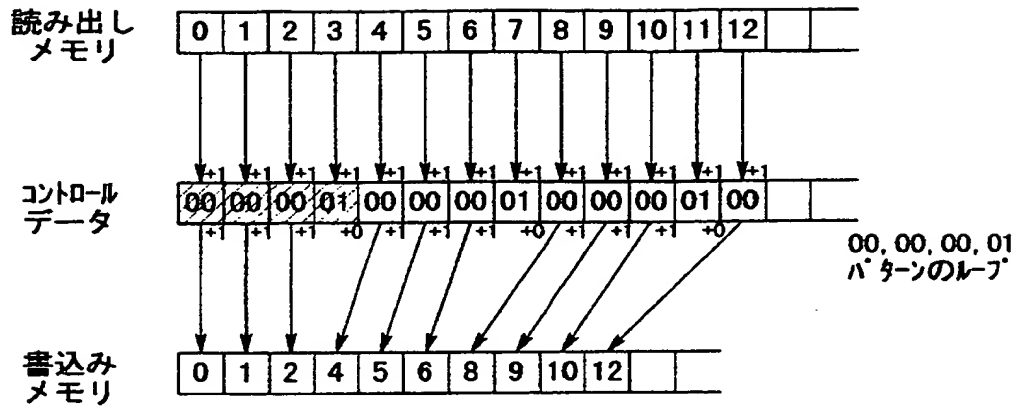
【図20】



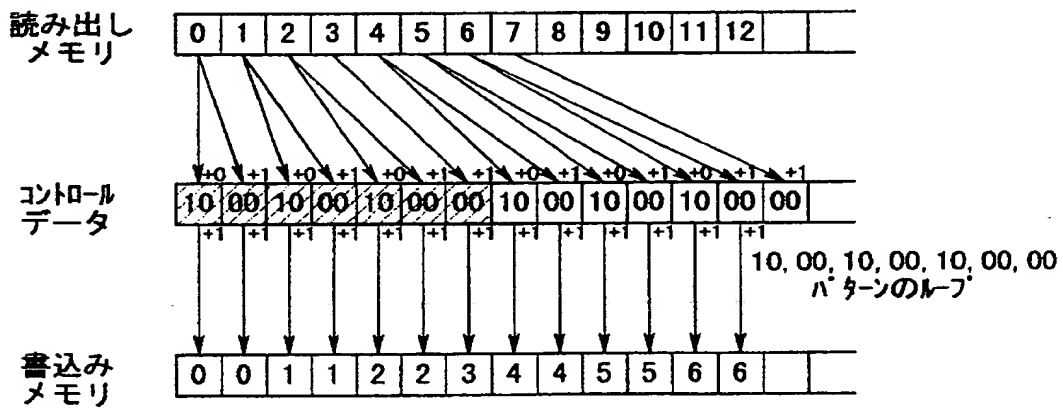
【図21】



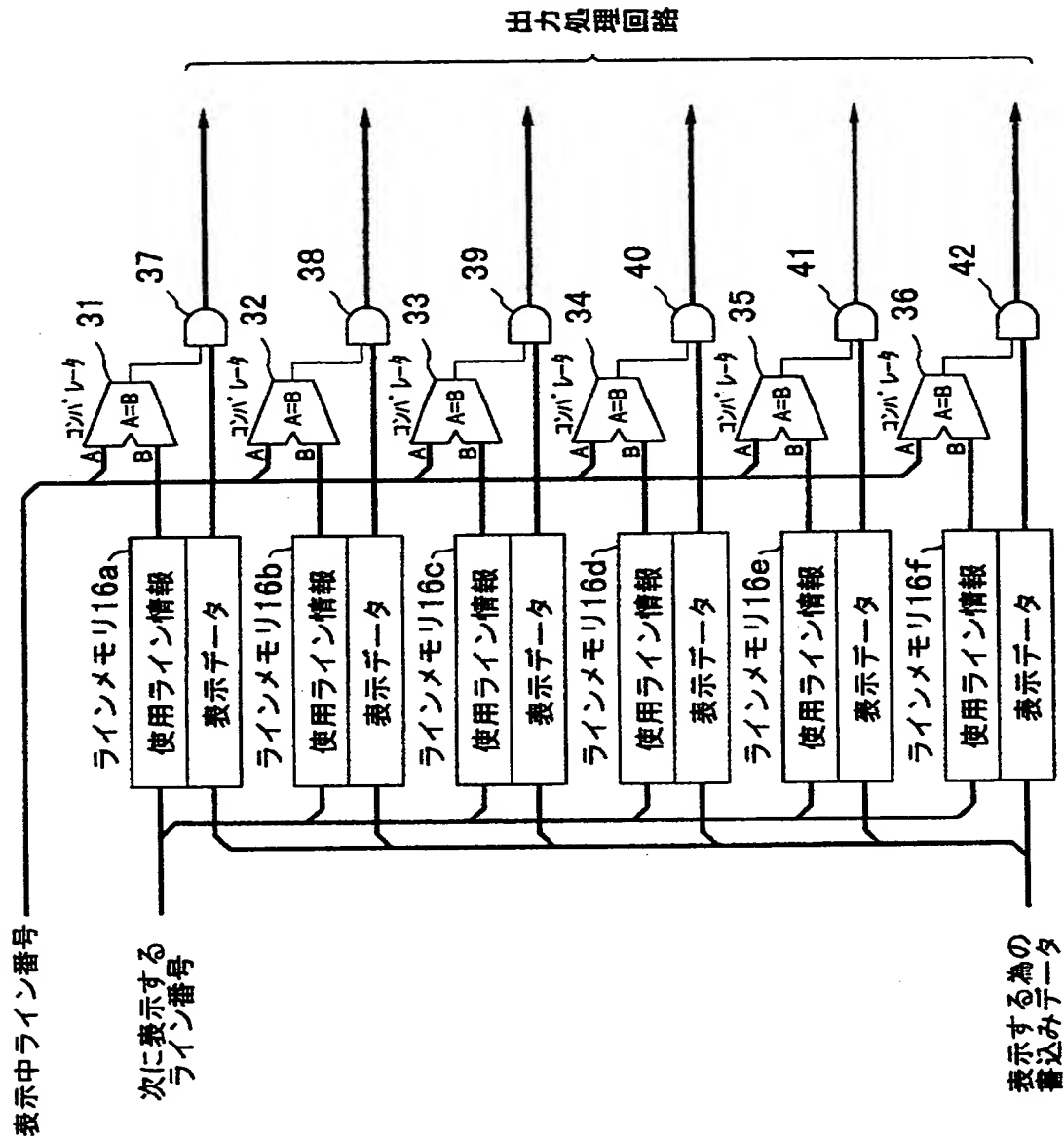
【図22】



【図23】

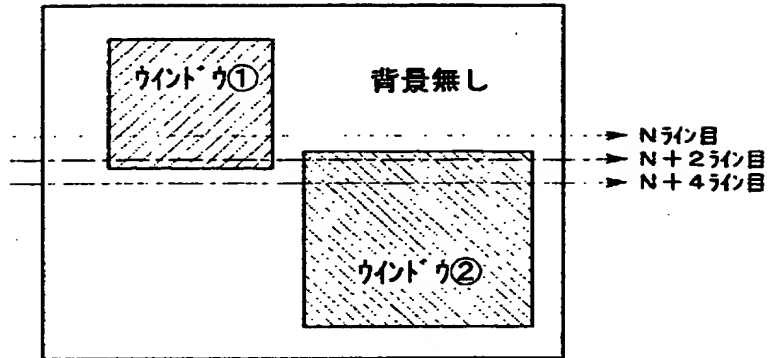


【図24】



【図 25】

(A)



(B)

| | | | | |
|-------------------|---------|---|-----------|---|
| ラインメモリ16a 内データ | 使用ライン情報 | 0 | N | 0 |
| | 表示データ | 0 | ウインドウ①データ | 0 |

↓ 使用ライン情報が(N)のデータのみ表示データ有効

| | | | |
|-------------|---|-----------|---|
| (N)ライン出力データ | 0 | ウインドウ①データ | 0 |
|-------------|---|-----------|---|

(C)

| | | | | | | |
|-------------------|---------|---|-----------|---|-----------|---|
| ラインメモリ16a 内データ | 使用ライン情報 | 0 | N+2 | 0 | N+2 | 0 |
| | 表示データ | 0 | ウインドウ①データ | 0 | ウインドウ②データ | 0 |

↓ 使用ライン情報が(N+2)のデータのみ表示データ有効

| | | | | | |
|---------------|---|-----------|---|-----------|---|
| (N+2)ライン出力データ | 0 | ウインドウ①データ | 0 | ウインドウ②データ | 0 |
|---------------|---|-----------|---|-----------|---|

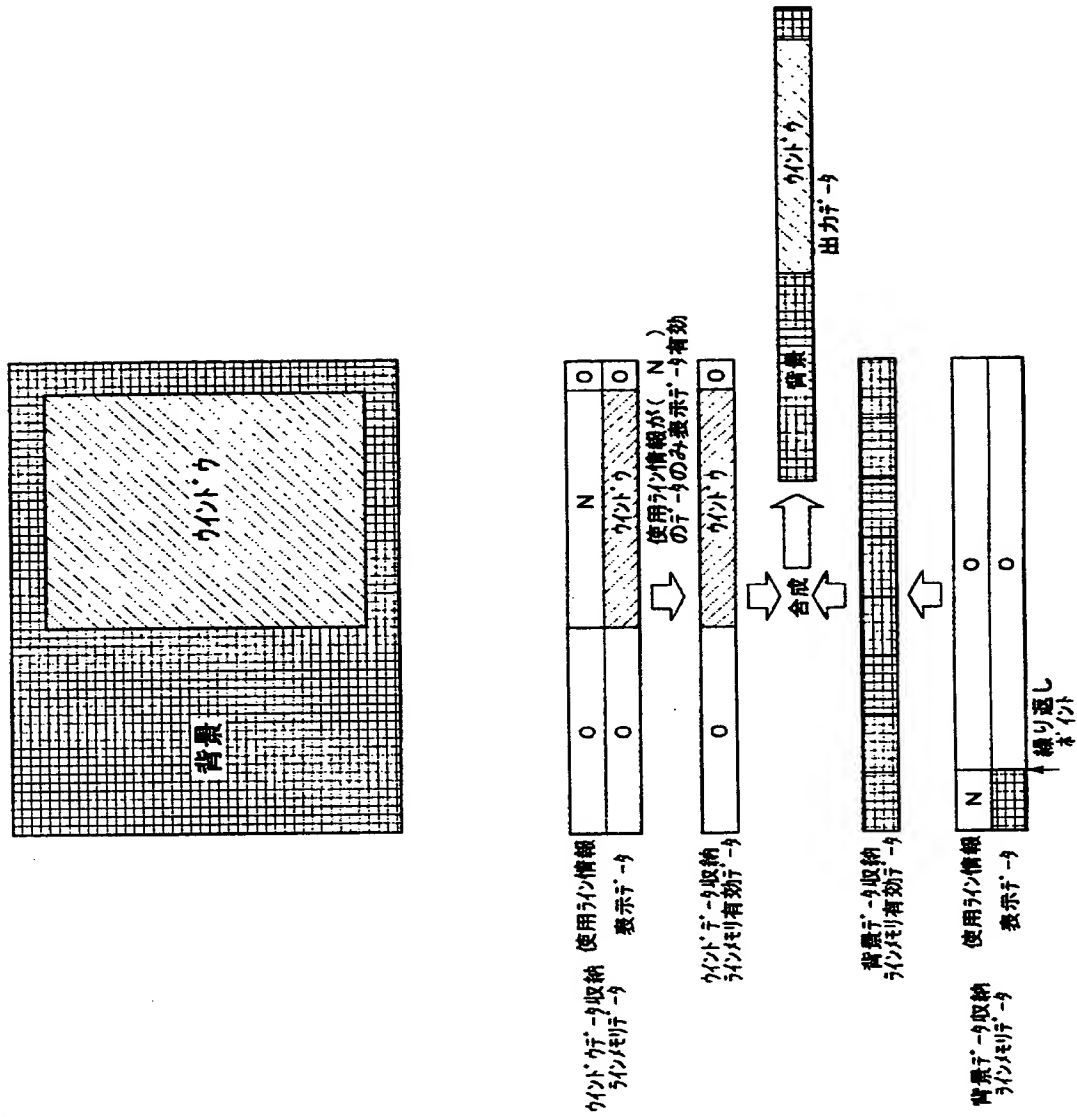
(D)

| | | | | | | |
|-------------------|---------|---|-----------|---|-----------|---|
| ラインメモリ16a 内データ | 使用ライン情報 | 0 | N+2 | 0 | N+4 | 0 |
| | 表示データ | 0 | ウインドウ①データ | 0 | ウインドウ②データ | 0 |

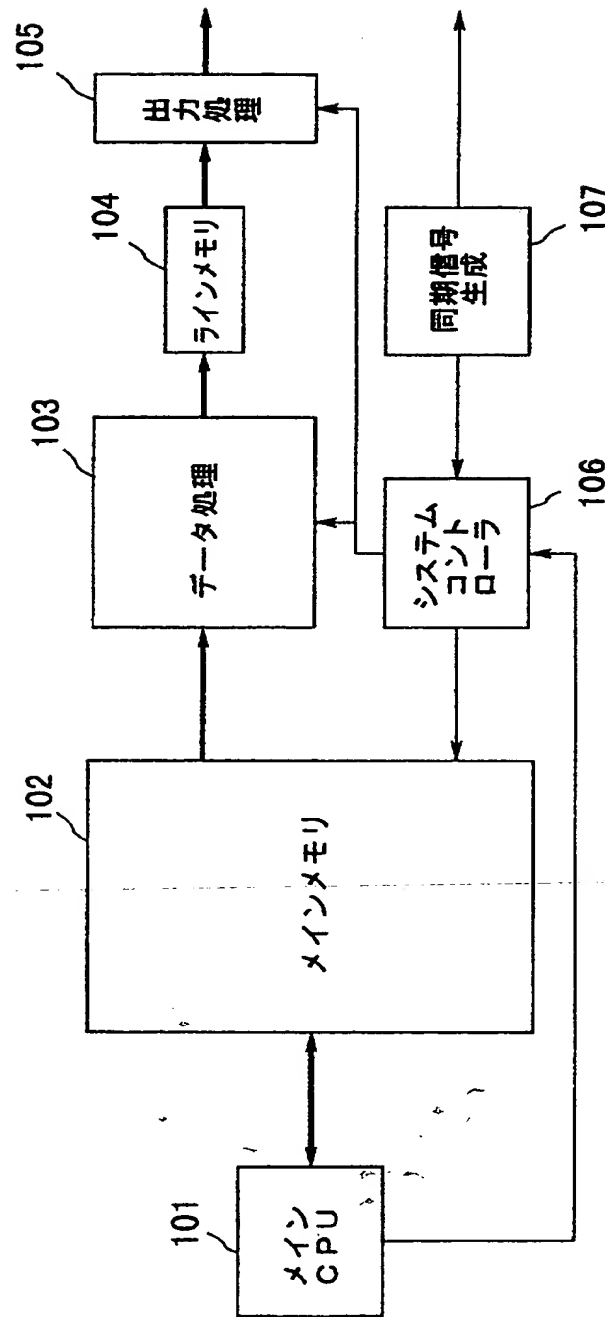
↓ 使用ライン情報が(N+4)のデータのみ表示データ有効

| | | | |
|---------------|---|-----------|---|
| (N+4)ライン出力データ | 0 | ウインドウ②データ | 0 |
|---------------|---|-----------|---|

【図26】



【図27】



【書類名】 要約書

【要約】

【課題】 表示データを収納するメモリ空間を必要なものだけとし、表示のためのメモリアクセス回数を抑えて処理を高速化できるとともに、主制御部の負担を軽減することができるプログラマブル表示装置を提供することである。

【解決手段】 メインCPU 11、プログラムや表示データやその他のデータを記憶するメインメモリ 12、メインメモリ 12の表示データをディスプレイ表示のデータ形式に変換する処理を行うデータ処理回路 13、変換処理された表示データを記憶する表示メモリ部 14、表示データを画面に出力するための処理を行う出力処理回路 17、メインメモリ 12へのデータアクセスを行うDMA 18、プログラムメモリ 19、データメモリ 20、プログラムメモリ 19やデータメモリ 20に記述された命令・データを解釈し、それに従っておもに表示データの転送等を行う表示プロセッサ 21、同期信号生成回路 22、とから構成される。

【選択図】 図 1

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000005049

【住所又は居所】 大阪府大阪市阿倍野区長池町22番22号

【氏名又は名称】 シャープ株式会社

【代理人】 申請人

【識別番号】 100069534

【住所又は居所】 東京都千代田区永田町2丁目14番2号 山王グラ
ンドビルディング3階317区 藤本特許法律事務所

【氏名又は名称】 藤本 博光

出 願 人 履 歴 情 報

識別番号 [000005049]

| | |
|----------|---------------------|
| 1. 変更年月日 | 1990年 8月29日 |
| [変更理由] | 新規登録 |
| 住 所 | 大阪府大阪市阿倍野区長池町22番22号 |
| 氏 名 | シャープ株式会社 |